

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
КАФЕДРА АВТОМАТИКИ ТА УПРАВЛІННЯ В ТЕХНІЧНИХ СИСТЕМАХ

«На правах рукопису»
УДК __004.4__

«До захисту допущено»

Завідувач кафедри

(підпис) О.І. Ролік
(ініціали, прізвище)

“ ____ ” _____ 20__ р.

Магістерська дисертація

зі спеціальності (спеціалізації) 121, інженерія програмного забезпечення
(інженерія програмного забезпечення комп'ютерних систем) _____
(код і назва спеціальності)

на тему: Система адміністрування електронних навчальних курсів _____

Виконав (-ла): студент (-ка) 2 курсу, групи ІТ-83мп _____
(шифр групи)

Романчук Сергій Анатолійович _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник доцент кафедри АУТС, к.т.н, доцент Полторак В.П. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант _____
(назва розділу) (науковий ступінь, вчене звання, прізвище, ініціали) (підпис)

Рецензент в.о. зав. каф. СПіСКС, д.т.н, доцент Романкевич В. О. _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2019 року

**Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»**

Факультет (інститут) Інформатики та обчислювальної техніки _____
(повна назва)

Кафедра Автоматики та управління в технічних системах _____
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною (освітньо-професійною) програмою

Спеціальність (спеціалізація) 121 інженерія програмного забезпечення
комп'ютерних систем _____
(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О. І. Ролік
(підпис) (ініціали, прізвище)

«___» _____ 20__ р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Романчуку Сергію Анатолійовичу _____
(прізвище, ім'я, по батькові)

1. Тема дисертації «Система адміністрування електронних навчальних курсів»

науковий керівник дисертації Полторак Вадим Петрович, к.т.н, доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «___» _____ 20__ р. № _____

2. Строк подання студентом дисертації _____

3. Об'єкт дослідження поліпшення взаємодії викладачів із електронними навчальними курсами, шляхом використання уніфікованої панелі адміністрування з можливістю розширення шляхом інтеграції сторонніх рішень, що забезпечують визначені функції, поліпшення процесу аналізу дій слухачів

4. Предмет дослідження (вихідні дані для магістерської дисертації за освітньо-професійною програмою) процес адміністрування електронних навчальних курсів шляхом їх створення, редагування, надання доступів, керування слухачами, імпорту курсів сторонніх розробників і автоматизації рутинних дій викладача.

5. Перелік завдань, які потрібно розробити проаналізувати існуючі рішення та виявити позитивні та негативні сторони, сформулювати вимоги до системи та сценарії використання, розробити структурну схему системи, розробити архітектуру системи, розробити інтерфейс взаємодії з системою, розробити стартап-проект

6. Орієнтовний перелік ілюстративного (графічного) матеріалу діаграма прецедентів, структурна схема системи, інфологічна модель БД, діаграма БД, діаграма розгортання, архітектура бібліотеки для роботи з Excel, загальна архітектура системи, діаграма послідовності імпорту стороннього курсу

7. Орієнтовний перелік публікацій Аналіз особливостей державних стандартів ЕЦП на властивостях еліптичних кривих _____

8. Консультанти розділів дисертації*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання 24 жовтня 2018

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Порівняльний аналіз існуючих рішень	02.09.19 – 06.09.19	
	Формування вимог до системи	09.09.19 – 13.09.19	
	Розроблення сценаріїв використання системи	16.09.19 – 20.09.19	
	Вибір та обґрунтування елементів та технологій	23.09.19 – 27.09.19	
	Розроблення структурної схеми системи	30.09.19 – 04.10.19	
	Розроблення ER-діаграми	07.10.19 – 11.10.19	
	Реалізація бізнес-логіки системи	14.10.19 – 18.10.19	
	Розроблення інтерфейсів взаємодії	21.10.19 – 25.10.19	
	Розроблення стартап-проекту	28.10.19 – 02.11.19	
	Оформлення графічного матеріалу	04.11.19 – 25.11.19	
	Оформлення текстового матеріалу	26.11.19 – 02.12.19	
	Подача дисертації на перевірку	03.12.19 – 18.12.19	

Студент

_____ (підпис)

С.А. Романчук

_____ (ініціали, прізвище)

Науковий керівник дисертації

_____ (підпис)

В.П. Полторак

_____ (ініціали, прізвище)

* Консультантом не може бути зазначено наукового керівника

РЕФЕРАТ

Магістерська дисертація містить 123 сторінок пояснювальної записки, 42 рисунки, 46 таблиць, 8 креслеників та 19 посилань на використані літературні джерела.

Актуальність дисертації полягає в можливості зміни або покращення процесу взаємодії викладача з електронними навчальними курсами.

Мета дисертації – розробка системи адміністрування електронних навчальних курсів.

Об'єктом дослідження є поліпшення взаємодії викладачів із електронними навчальними курсами, шляхом використання уніфікованої панелі адміністрування з можливістю розширення шляхом інтеграції сторонніх рішень, що забезпечують визначені функції, поліпшення процесу аналізу дій слухачів.

Предметом дослідження є процес адміністрування електронних навчальних курсів шляхом їх створення, редагування, надання доступів, керування слухачами, імпорту курсів сторонніх розробників і автоматизації рутинних дій викладача. Зокрема, процес оцінювання слухача шляхом використання диференційованої оцінки.

Методами дослідження є емпіричне дослідження, теоретичне дослідження, абстрагування і конкретизація.

В результаті виконання робіт пов'язаних із магістерською дисертацією розроблено систему адміністрування електронних навчальних курсів та опубліковано статтю на тему «Аналіз особливостей державних стандартів ЕЦП на властивостях еліптичних кривих».

Ключові слова: електронний курс, система адміністрування, веб-застосування, API, імпорт, експорт, взаємодія з сторонніми системами.

SUMMARY

The master's thesis contains 123 pages of explanatory note, 42 images, 46 tables, 8 drawings and 19 references to used literature sources.

The relevance of the dissertation is the possibility of changing or improving the process of teacher interaction with e-courses.

The purpose of the dissertation is to develop the system of administration of electronic training courses.

The object of the study is to improve the interaction of teachers with e-courses through the use of a unified extension panel by integrating third-party solutions that provide certain features and improve the process of analyzing student actions.

The subject of the study is the process of administering e-courses by creating, editing, providing access, managing students, importing third-party courses, and automating the teacher's routine activities. In particular, the process of evaluating the listener by using differentiated assessment.

Research methods are empirical research, theoretical research, abstraction and concretization.

As a result of the work related to the master's thesis, the system of administration of electronic training courses was developed and an article was published on the topic "National EDS standards properties analysis".

Keywords: e-course, administration system, web application, API, import, export, interaction with third-party systems.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ	8
ВСТУП.....	9
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	12
1.1 Система управління комплексом лабораторних робіт з криптографії на ЕК	12
1.2 Система управління навчальним процесом «Moodle».....	14
1.3 Платформа для дистанційного навчання «Prometheus»	16
1.4 Платформа для онлайн-навчання «Coursera»	17
1.5 Таблиця порівняння існуючих рішень	18
2. ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ.....	22
3. РОЗРОБЛЕННЯ СЦЕНАРІЇВ ВИКОРИСТАННЯ СИСТЕМИ.....	37
3.1 Актори та залежності сценаріїв використання	37
3.2 Опис сценаріїв використання	41
4. ВИБІР ТА ОБГРУНТУВАННЯ ЕЛЕМЕНТІВ ТА ТЕХНОЛОГІЙ	47
4.1 Огляд варіантів реалізації	47
4.2 Вибір бази даних та сховища.....	47
4.3 Вибір технологій для серверного застосування	51
4.4 Вибір технологій для реалізації веб-застосування.....	53
5. РОЗРОБЛЕННЯ СТРУКТУРНОЇ СХЕМИ СИСТЕМИ	54
5.1 Загальна структурна схема	54
5.2 Детальний огляд структурних елементів	57
6. РОЗРОБЛЕННЯ ER-ДІАГРАМИ	60
6.1 Інфологічна модель	60
6.2 Модель сутність-зв'язок.....	64

6.3	Опис бази даних	67
7.	РЕАЛІЗАЦІЯ БІЗНЕС-ЛОГІКИ СИСТЕМИ	74
7.1	Вибір підходу та загальна архітектура системи	74
7.2	Розробка правил валідації	81
8.	РОЗРОБЛЕННЯ ІНТЕРФЕЙСІВ ВЗАЄМОДІЇ	86
8.1	API та документація	86
8.2	Інтерфейс користувача	90
9.	РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ	95
9.1	Опис ідеї проекту	95
9.2	Концепції конкурентів	97
9.3	Технічний аудит ідеї проекту	98
9.4	Аналіз ринкових можливостей запуску проекту	99
9.4.1	Попередня характеристика потенційного ринку стартап проекту	99
9.4.2	Характеристика потенційних клієнтів стартап проекту	100
9.5	Фактори загроз	101
9.6	Фактори можливостей	104
9.7	Ступеневий аналіз конкуренції на ринку	105
9.8	Аналіз конкуренції в галузі за м. Портером	107
9.9	Обґрунтування факторів конкурентоспроможності	110
9.10	Порівняльний аналіз сильних та слабких сторін	112
9.11	Swot аналіз стартап-проекту	113
9.12	Розроблення маркетингової програми стартап-проекту	114
	ВИСНОВКИ	119
	ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	121

ПЕРЕЛІК СКОРОЧЕНЬ ТА ТЕРМІНІВ

API – application programming interface.

ER – entity-relationships.

БД – база даних.

ВС – відкрита система.

ЕК – еліптична крива.

ЖЦ – життєвий цикл.

ІС – інформаційна система.

ІТ – іноформаційна технологія.

ОС – операційна система.

САЕНК – система адміністрування електронних навчальних курсів.

Платформа - це сервіс, що надає користувачеві готовий до роботи рушій і дозволяє керувати процесом без використання додаткових засобів, може бути використаний як основа для розширення та кастомізації.

ВСТУП

Наслідком бурхливого розвитку інформатизації суспільства, постійного зниження вартості послуг на використання глобальної мережі Internet та суттєве поглиблення процесів упровадження інформаційних технологій в освітню практику стала поява дистанційного навчання як найбільш перспективної, гуманістичної, інтегральної і орієнтованої на індивідуалізацію форми освітнього процесу. Не дивлячись на те, що до інформації неможливо доторкнутися, побачити її колір, відчутти її запах, тощо, вона потребує захисту від багатьох загроз [1].

Широкий доступ студентів до освітніх ресурсів, не обмежені часом та відстанню можливості працювати з електронними варіантами програмно-методичного забезпечення по тим чи іншим дисциплінам формують відповідний рівень предметного діалогу «викладач-студент». У зв'язку із цим, новою функцією викладача є пошук діалогових комунікацій, що підвищують мотивацію навчання, сприяють розвитку пізнавальних інтересів студентів [2].

Проблема впровадження багатьох орієнтованих на конкретний предмет електронних лабораторних практикумів пов'язана з відсутністю можливості контролювати навчальний процес із однієї інформаційної системи, часто редагування матеріалів, розширення бази запитань для оцінювання неможливе. Під інформаційною розумітимемо будь-яку систему, яка за допомогою технічних засобів виконує одну або кілька таких функцій, як збирання, передавання, перетворення, накопичення, зберігання та оброблення інформації [1]. Вагомим фактором мотивації також є можливість провалу та перездачі – це зменшує тиск і нервові напруження слухача, що проходить оцінювання та дозволяє отримати ще більш об'єктивний результат.

Актуальність роботи полягає в можливості зміни або покращення процесу взаємодії викладача з електронними навчальними курсами. Оскільки проблема наведена вище досі не є вирішеною повністю, потребується дослідження та розробка системи, яка покращить існуючі засоби оцінювання знань, дозволить контролювати не лише оцінку за завдання а й активність та зацікавленість слухача, а також буде

враховувати це під час виставлення кінцевої оцінки. З іншого боку, система дозволить значно зекономити час викладача на рутинній роботі – таким чином вирішивши проблему контролю над кількома комплексами.

На сьогодні частковим вирішенням проблеми уніфікації інтерфейсу для багатьох курсів займається Open Source співтовариство розробників, які працюють над платформою Moodle. Окрім них проблемою впровадження електронних систем у навчальних процес зайнявся Кабінет Міністрів України шляхом фінансування та оцінки стартап проектів.

Предметом дослідження є процес адміністрування електронних навчальних курсів шляхом їх створення, редагування, надання доступів, керування слухачами, імпорту курсів сторонніх розробників і автоматизації рутинних дій викладача. Зокрема, процес оцінювання слухача шляхом використання диференційованої оцінки.

Об'єктом дослідження є поліпшення взаємодії викладачів із електронними навчальними курсами, шляхом використання уніфікованої панелі адміністрування з можливістю розширення шляхом інтеграції сторонніх рішень, що забезпечують визначені функції, поліпшення процесу аналізу дій слухачів.

Розробка системи адміністрування електронних навчальних курсів сприятиме впровадження нових електронних лабораторних практикумів, організація роботи з якими буде ефективнішою та забезпечить більш об'єктивне оцінювання студентів. Запровадження в навчальний процес нових технологій навчання значно підвищує відповідальність викладача за якість навчально-методичних матеріалів, зміст яких потребує постійного поновлення і вдосконалення, а досвід традиційної організації навчального процесу – адаптації до нової системи комунікацій між викладачем і студентом. Застосування автоматизованих систем контролю знань, консультування і навчання за допомогою комп'ютерних технологій дозволяє істотно скоротити витрати часу викладача на етапах контролю знань, зняти перенапругу та нервозність студентів в процесі навчання та складання іспитів, зробити процес навчання більш яскравим і привабливим.

Метою дослідження є проектування та розробка першої версії системи адміністрування електронних навчальних курсів, яка вирішить проблему керування кількома розрізненими лабораторними практикумами а також підтримуватиме можливість створення вбудованих простих електронних курсів.

Відповідно до зазначеної мети поставлено такі завдання:

- проаналізувати існуючі рішення та визначити критерії для покращення;
- визначити та розробити варіанти використання системи відповідно до типу користувача;
- проаналізувати існуючі технології та вибрати підходящі для розробки системи;
- визначити структурні елементи майбутньої системи та розробити структурну схему;
- спроектувати базу даних і архітектуру системи;
- реалізувати API для методів та функцій системи;
- розробити стартап-проект.

Під час виконання робіт використовуються такі методи досліджень:

- методи емпіричного дослідження;
- методи теоретичного дослідження;
- загальні методи (абстрагування і конкретизація).

Підвищення ефективності та якості навчального процесу з комп'ютерної криптографії неможливе без лабораторного практикуму, який дає можливість не тільки перевірити деякі теоретичні положення курсу, але й формує та удосконалює експериментальні уміння та навички студентів. Зміст лабораторного практикуму повинен відповідати навчальній та робочій програмі дисципліни, відображати і експериментально розкривати основні положення теми. Довгий час криптографія в основному застосовувалася для забезпечення секретності листування в дипломатії, військовій справі, спецслужбами. З розвитком інформаційних технологій застосування криптографічних методів стало актуальним для всього суспільства.

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Система управління комплексом лабораторних робіт з криптографії на ЕК

Розроблена студентами система управління лабораторними роботами із криптографії на еліптичних кривих (рис. 1) більше не задовольняє вимоги сучасного студента та викладача. Доступ до завдань можливий тільки з-під ОС Windows, крім того прослідковуються проблеми з сумісністю програмного застосування на нових версіях системи, якими здебільшого обладнані комп'ютери користувачів. На ОС Windows 10, версія 1903 – програма не запускається навіть у режимі сумісності з ранніми версіями ОС.

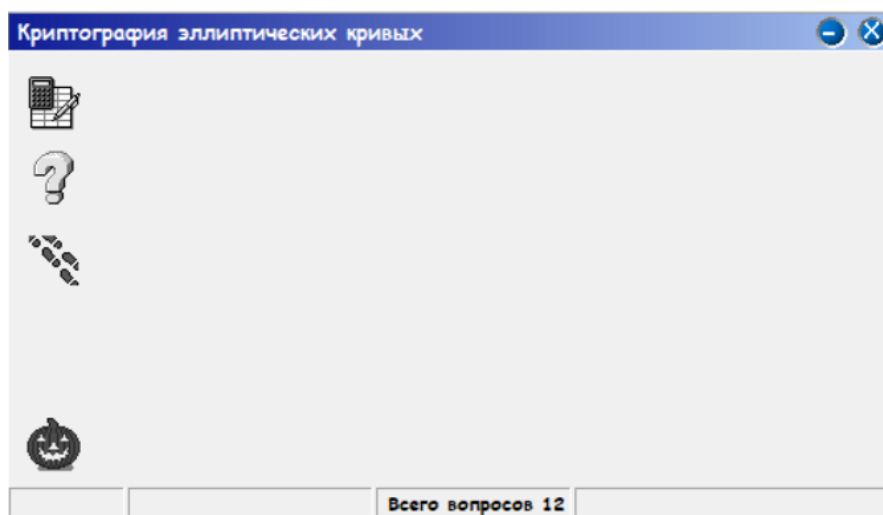


Рисунок 1 – Загальний вигляд програми тестування знань

Наведена система управління комплексом лабораторних робіт дозволяє провести перевірку знань і ознайомити студентів з теоретичними засадами. Оцінювання відбувається у формі тестових питань та завдань з можливістю введення відповіді вручну. Зміна інформації відбувається шляхом відкриття діалогових вікон (рис. 2). Окрім наведених недоліків процес внесення змін до курсу є складним та незручним для викладача. Очевидно, що матеріали додані в систему оновлюються рідко. У випадку внесення змін усі студенти повинні перевстановити застосування на своїх комп'ютерах (з втратою даних).

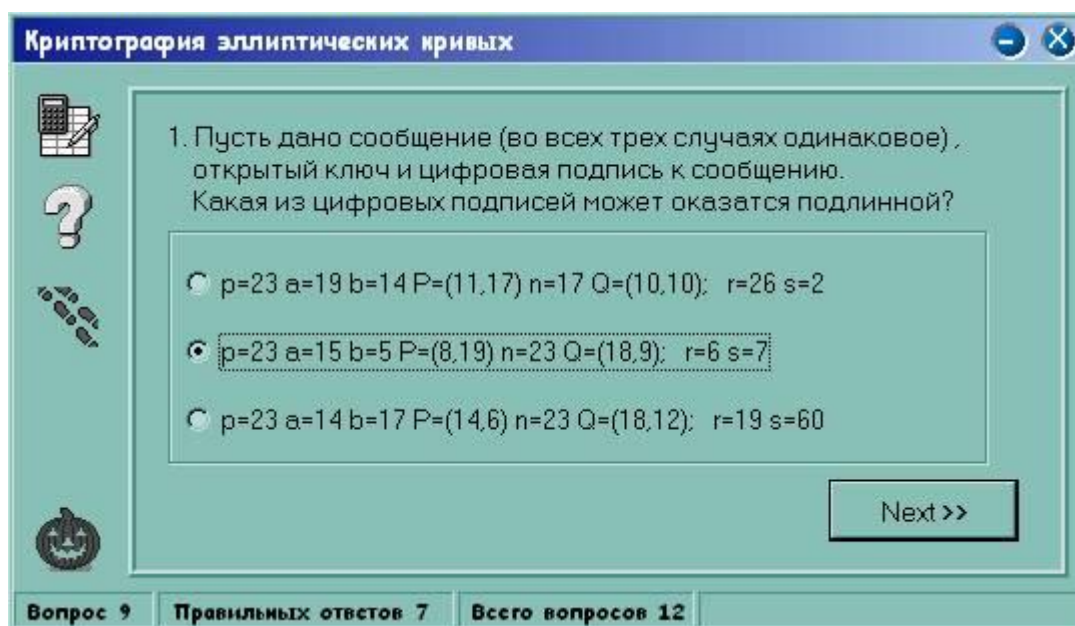


Рисунок 2 – Процесс тестування

Окрім орієнтованих на студента функцій система дає можливість познайомитися з історією запусків користувача (рис. 3), проте не забезпечує викладача функціональністю перегляду детальних результатів і статистики користувача, що є важливим при оцінюванні знань.

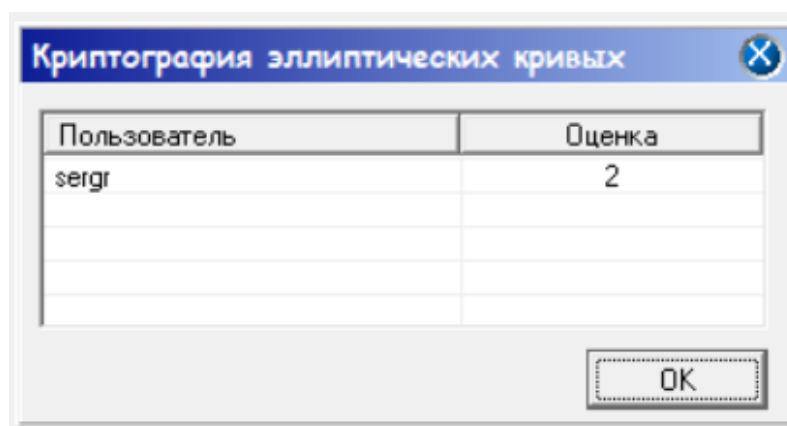


Рисунок 3 – Результати оцінювання користувача

Суттєвим недоліком наведеного програмного комплексу є спосіб його розповсюдження. Інсталяція системи відбувається шляхом копіювання застосунка з мережевої теки на комп'ютер користувача. Таким чином недобросовісні студенти

можуть змінити справжні результати оцінювання через використання програм ArtMoney[3] або CheatEngine[4] шляхом підміни значення змінної у пам'яті в процесі проходження тестування. Крім того, система не забезпечує достатньо тісної співпраці викладача зі студентом і не може бути використана при впровадженні дистанційного навчання, оскільки процес оцінювання знань повинен чітко контролюватися на предмет взлому системи.

1.2 Система управління навчальним процесом «Moodle»

Навчальна платформа «Moodle» призначена для створення надійної навчальної системи, в яку увійдуть студенти, батьки, викладачі, менеджери, тощо. Охоплення аудиторії даного комплексу на 2019 рік складає 229 країн, а інтерфейс (рис. 4) підтримує кілька десятків мов. За останні 5 років система знайшла підтримку та користувачів у 32 країнах [5]. Такий результат в першу чергу пов'язаний з тим, що цей проект підтримуються Open Source спільнотою, програмний код платформи є у вільному доступі. Крім того, розробники передбачили можливість розширення через встановлення додаткових модулів.

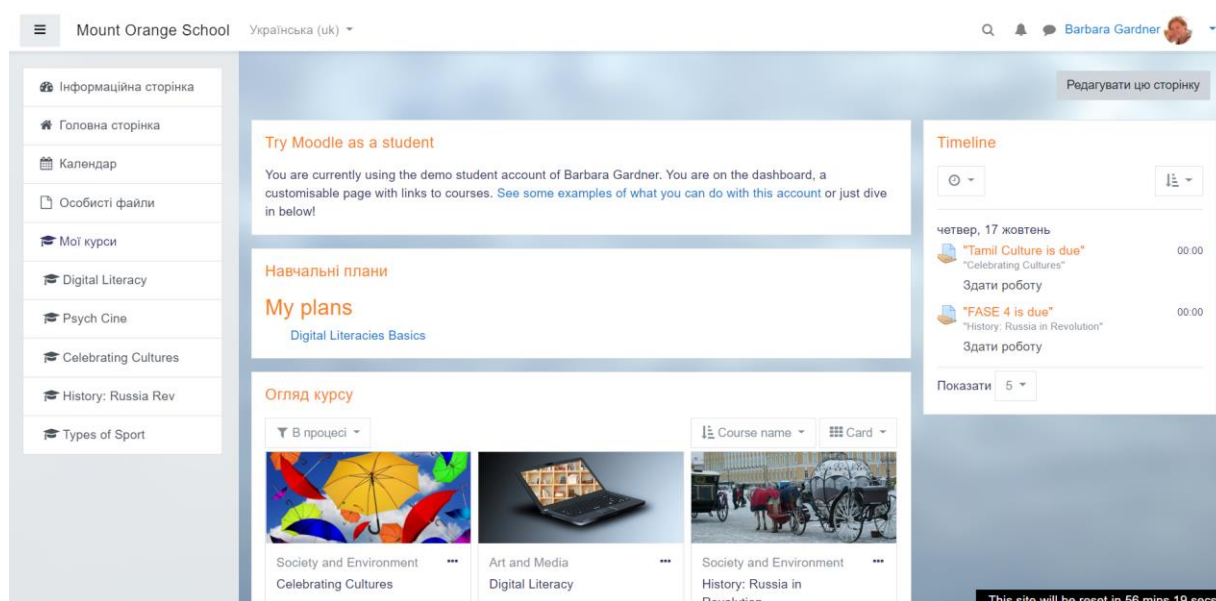


Рисунок 4 – Інтерфейс системи «Moodle»

Платформа містить дуже потужні інструменти для об'єднання викладача та слухача, зокрема:

- форум, для дискусій;
- приватні повідомлення;
- нотатки;
- блоги;
- календар.

Все це забезпечує тісний зв'язок та взаємодію під час проведення навчального курсу, тому платформа використовується для дистанційного та змішаного типів навчання.

Для викладача розроблено зручний кабінет моніторингу курсів, результатів (рис. 5) та активностей слухачів.

The screenshot shows the 'Журнал оцінок' (Gradebook) for the 'Celebrating Cultures' course. It displays a list of participants with their names, email addresses, and scores for two workshops. The interface is in Ukrainian.

Ім'я / Прізвище	Електронна пошта	Workshop: My home cou...	Workshop: My home cou...	Database: Food I
Frances Banks	francesbanks231@example.com	26,40	18,22	
Angela Bowman	angelabowman379@example.com	59,60	16,44	
Lao Cai	laocai154@example.com	80,00	18,22	
Paul Castillo	paulcastill270@example.com	19,80	18,22	

Рисунок 5 – Сторінка моніторингу оцінок

Журнал оцінок може бути відредагований викладачем вручну. Ця можливість зручна для випадків очної перездачі робіт, технічних збоїв, нарахування додаткових балів, тощо.

Окрім оцінювання функціональність платформи включає такі базові дії:

- теоретичні відомості з можливістю обговорення на форумі;

- дискусії з участю викладача та інших слухачів;
- новини та анонси подій, зокрема відкриття курсів, з можливістю розділення для груп, окремих користувачів, слухачів певних курсів;
- онлайн тестування;
- стилізація середовища на власний смак;
- завантаження файлів та можливість поширення їх для вибраних груп, осіб;
- впровадження сторонніх модулів.

Окрім наведених дій, система забезпечується бібліотекою, що дозволяє конвертувати сторінки в PDF.

1.3 Платформа для дистанційного навчання «Prometheus»

Найбільший проект для безкоштовної освіти в Україні (рис. 6) [6]. Містить сотні онлайн курсів, позиціонується, як платформа для дистанційного навчання.

Для зручності взаємодії також має форум для обговорень між викладачем та слухачами. Подача матеріалу орієнтована на відео-лекції та презентації. Оцінювання знань відбувається шляхом проходження тестування та творчих вправ. По закінченню курсу видається сертифікат.

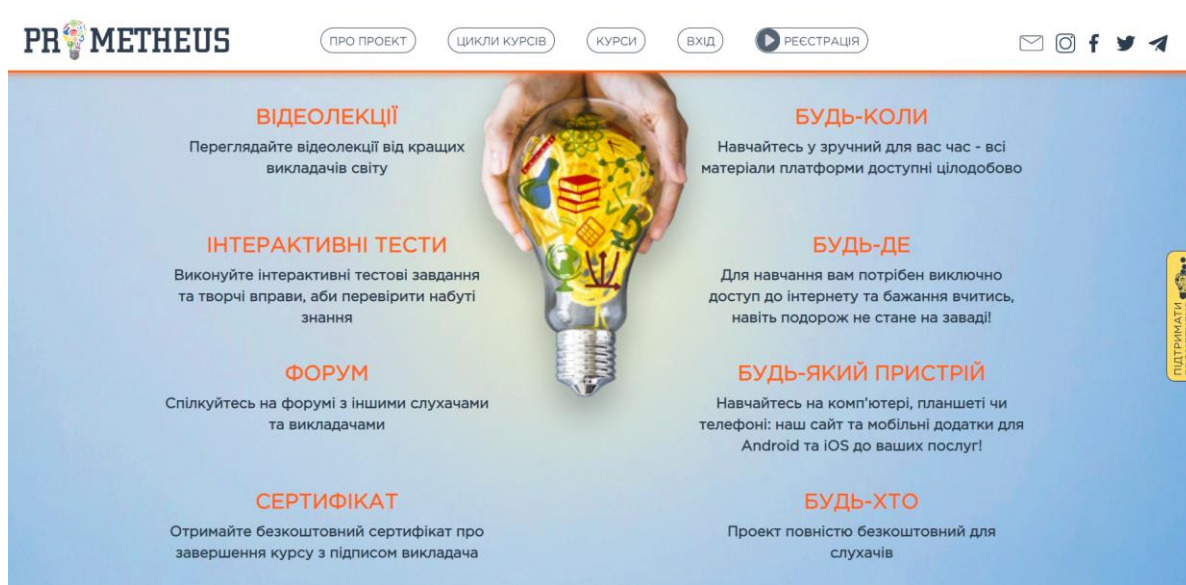


Рисунок 6 – Головна сторінка «Prometheus»

Розробники платформи ставлять за мету наскрізне впровадження змішаної освіти. Проект розвивається дуже активно, постійно поповнюється новими курсами та впроваджуються сучасні навчальні методи.

До недоліків даної платформи можна віднести те, що запис на курс проводиться раніше ніж дата запуску курсу. Немає можливості записатися на курс, що розпочався. Важливо виділити те, що курси та матеріали подаються українською.

1.4 Платформа для онлайн-навчання «Coursera»

Найпопулярніша в світі платформа онлайн-навчання, забезпечує доступ до тисяч відкритих курсів від викладачів з усього світу, займає перші та високі місця у різних рейтингах [7], з якісно продуманою функціональністю та інтерактивною складовою. Курс даної платформи орієнтовано на слухача, він містить текстові лекції, відеоматеріали, навчальні проекти та тести (рис.7). Для комунікації між слухачами платформа включає в себе форум для обговорення матеріалів курсу, поширення ідей та допомоги в засвоєнні матеріалу.

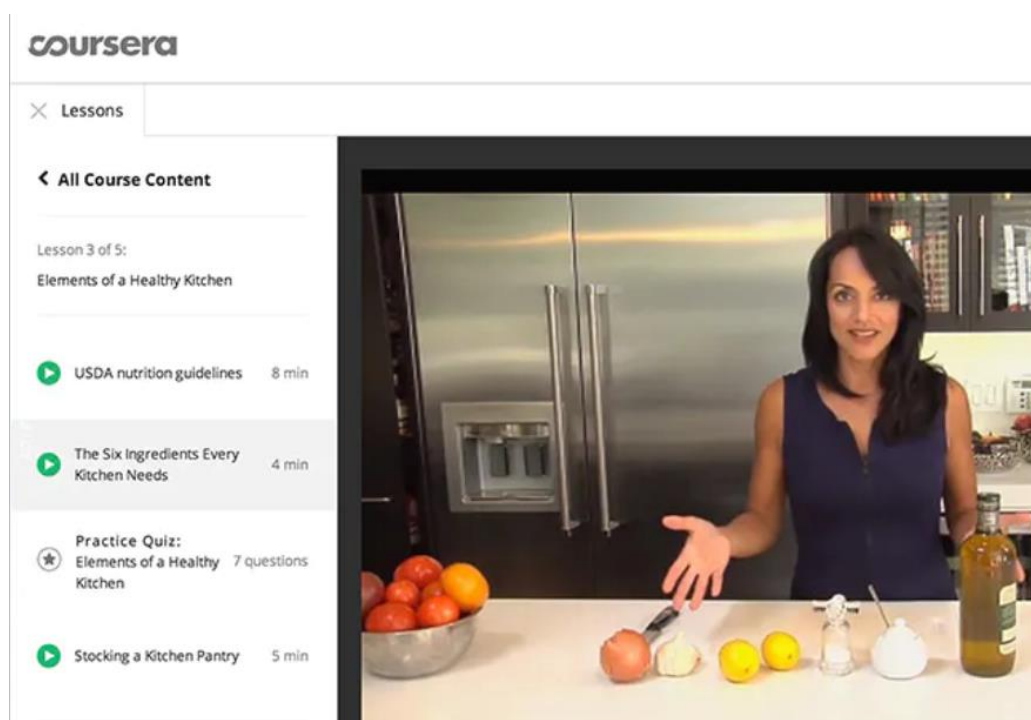


Рисунок 7 – Курс на платформі «Coursera»

Після завершення курсу слухачу видається онлайн сертифікат. Навідміну від інших через наскрізну розпоширеність по світу сертифікати платформи з легкістю імпортуються до всеможливих онлайн-сервісів пошуку роботи та фріланс-біржі.

Це часто стає одним із ключових критеріїв при виборі онлайн-платформи навчання. Не всі курси та матеріали є безкоштовними. Найбільші розчарування виникають у слухачів, що розраховували на безкоштовний курс, але по проходженню 80% курсу вони виявляють, що частина лекцій та тестів (необхідні для отримання сертифікату) платні.

1.5 Таблиця порівняння існуючих рішень

Порівняння існуючих рішень проведено з метою вибору критеріїв для покращення через розробку нової системи. Під час вибору пунктів акцент зроблено на можливості інтеграції в навчальний процес університету для студентів денної форми навчання; функціональності, що забезпечує контроль через журнал оцінок та детальну статистику; операціях, що повинні забирати мінімальну кількість часу викладача.

Для порівняння вибрано наступні критерії (табл. 1.1):

- сертифікат для слухача. Вибрані аналоги нової системи функціонують як самостійні продукти, що не впроваджуються до навчального процесу, тому результати проходження курсів підтверджуються власними електронними сертифікатами;
- інтерактивне навчання. Продукти що порівнюються забезпечують слухача наглядними матеріалами, при необхідності проектними або сторонніми рішеннями для більш детального вивчення теми;
- тестування. Як один з найпоширеніших методів оцінювання знань, включає в себе завдання з вибором правильної відповіді серед кількох запропонованих;

- лекційні матеріали. Слухач забезпечується усіма мінімально необхідними теоретичними та довідковими матеріалами;
- сторонні довідкові матеріали. Слухач має доступ до систематизованих посилань на сторонні ресурси для більш детального вивчення запропонованої тематики;
- безкоштовний доступ. Відкритість для широкого кола слухачів можлива за умови безоплатного доступу до матеріалів курсів у межах встановлених навчальним закладом або викладачем;
- можливість інтеграції в навчальний процес університету. Система може бути інтегрована викладачем у навчальний процес вибраного предмету для студентів денної форми навчання, використовуватися в якості доповнення;
- моніторинг результатів. Викладач, як активний учасник навчального процесу при очній або змішаній формі навчання відповідає за якість знань слухачів. Можливість постійного та детального контролю за активністю та оцінками студентів є важливим чинником під час фінального оцінювання;
- тип реєстрації. Для отримання доступу до системи або окремих курсів слухач повинен пройти реєстрацію, може проводитися викладачем для вибраних груп або бути особистою (в т.ч. із застосуванням сторонніх сервісів).

Таблиця 1.1 – Порівняння існуючих рішень та вибір критеріїв для покращення

Критерій	«Криптографія на ЕК»	Moodle	Prometheus	Coursera
Сертифікат для слухача	Немає, результат враховується в оцінку за предмет	Є	Є	Є, можливий імпорт до сторонніх сервісів
Тестування	Є	Є	Є	Є
Інтерактивне навчання	Частково, забезпечує тільки теорією, тестом та	Є, забезпечується через сторонні модулі та	Є, відео-курси, форум для спілкування	Є, інтерактивний посібник, відео-курси,

Критерій	«Криптографія на ЕК»	Moodle	Prometheus	Coursera
	імітацією наочного відображення процесу	тісний зв'язок між усіма учасниками навчального процесу	між учасниками та проектні завдання	проектні роботи, тестування, взаємодія між слухачами
Лекційні матеріали	Тільки вибрані довідкові матеріали в HTML форматі	Можливість поширення файлів з лекціями	Відео-матеріали, підручники, посилання	Всі необхідні для успішного завершення курсу
Безкоштовний доступ	Є	Є	Є, повністю безоплатний	Частково, містить платні теми, курси
Можливість інтеграції в навчальний процес університету	Зараз інтегрований	Зараз інтегрований	В якості доповнення до основних лекцій	В якості доповнення до основних лабораторних
Моніторинг результатів	Історія результатів	Журнал оцінок, кращі та гірші слухачі	Сертифікати, відгуки, форум	Сертифікати, форум
Тип реєстрації	Немає	Викладач або адміністратор забезпечує слухачів обліковими записами	Особиста	Особиста

На основі проведеного аналізу визначено пункти, які є стандартом для вибраних продуктів, та ті, що можуть бути вдосконалені в новій системі. Для цього вибрано орієнтацію на такі критерії:

- повна відкритість у кожній з наведених систем;
- частково платна функціональність, що може стати відкритою;
- може бути автоматизоване за рахунок скорочення кількості необхідних дій з боку користувача (викладача або слухача);
- функціональність зарекомендувала себе з якісного боку – має позитивні відгуки від слухачів та викладачів.

Першим пунктом, який можна вдосконалити вибрано надання доступу до системи – реєстрацію. Система повинна підтримувати реєстрацію слухачів одиночну та масову. Реєстрація може бути особиста або викладачем для студента, групи, потоку, курсу. Наступним є те, що система має бути безкоштовною, проте може містити непрямі способи монетизації. Третім пунктом виділено можливість інтеграції в навчальний процес для студентів денної форми навчання. Система повинна забезпечувати функціональність оцінювання, подібну до існуючих продуктів – тестування, проекти, та видавати навчальний курс у формі лабораторних робіт або тем. Оскільки заплановано інтеграцію до навчального закладу викладач повинен бути активним учасником процесу та мати можливість детального моніторингу результатів слухачів. Лекційні матеріали можуть бути поширені через сторонні ресурси (Google Drive, Dropbox, шарингові системи), проте дані необхідні для вивчення теми мають бути систематизовані та доступні під час вивчення теоретичних матеріалів. Сучасне суспільство все більшою мірою спирається на інформаційні процеси (ІП), які стають рушійною силою економіки, суспільних відносин, військової справи. ІП - це процеси збору, підготовки, передачі, обробки, перетворення та використання інформації в різних сферах суспільства. Інформація, за одним із продуктивних визначень, є корисні, або ж, нові відомості про навколишній і внутрішній світ. Вона не дається нам безпосередньо, через органи відчуттів. Вона дана нам опосередковано, через фізичні носії - знаки, символи, сигнали (збурення фізичного стану середовища розповсюдження), тощо. Зазвичай їх називають даними [8]. Видача сертифіката необов'язкова, а мотивацією на поглинання запропонованої інформації є структурована подача та врахування результатів проходження електронного курсу в загальну оцінку.

2. ФОРМУВАННЯ ВИМОГ ДО СИСТЕМИ

На основі визначених раніше користувацьких критеріїв відбувається формування вимог до нової системи. Серед основних завдань захисту даних в ІС є забезпечення конфіденційності об'єктів інформаційної діяльності (ОІД) таких, як повідомлення, тексти, дані, документи, тощо; забезпечення аутентифікації ОІД та суб'єктів інформаційної діяльності (СІД); забезпечення механізмів і засобів виявлення порушень цілісності ОІД та СІД; забезпечення доступності ІС та ОІД для легальних користувачів, СІД; нонрепудіація - унеможливлення відмови СІД від виконаних в ІС дій, або прийнятих на себе обов'язків, має значення для електронного документообігу, більш точно - це унеможливлення відмови відправника від факту передачі повідомлення, а одержувача - від факту отримання повідомлення. Ці поняття мають свої визначення [8]. Даний розділ розділено на формування функціональних вимог, де описано очікувану поведінку системи та функції, що виконуються оформлені у формі системних вимог, що максимально докладно описують продукт та нефункціональні вимоги.

Тільки авторизовані користувачі мають доступ до системи. Аутентифікація користувача проводиться шляхом введення ним адреси електронної пошти та пароля. Користувачі системи, шляхом авторизації отримують визначену роль викладача або слухача з правами адміністрування, редагування, наповнення, відкриття та надання доступу до електронного навчального курсу та обмеженими правами читача, відповідно. Облікові записи викладачів налаштовуються через файл конфігурації системи, зміна облікових даних неможлива без перезапуску системи.

Створення облікового запису слухача можливе кількома способами:

- особиста реєстрація, через введення даних у відповідному вікні;
- одиночна реєстрація викладачем, вікно аналогічне до особистої реєстрації слухача, але доступне із панелі керування користувачами для користувача в ролі «Викладач»;
- масова реєстрація слухачів (з мінімальною кількістю дій з боку викладача).

Для інтеграції в навчальний заклад система має підтримувати групи. В контексті проекту група – сукупність користувачів у ролі «Студент», що об'єднані єдиним унікальним ідентифікатором нечутливим до регістру – кодом групи. Код групи може містити цифри 0-9, літери А-Я, та спеціальні символи.

Особиста реєстрація слухача та одиночна реєстрація викладачем відбувається шляхом вводу до системи наступних даних:

- адреса електронної пошти;
- повне ім'я;
- код групи.

Масова реєстрація реалізована шляхом імпорту до системи електронного файлу, який надається викладачу в електронному форматі старостою групи.

Файл для імпорту повинен відповідати встановленим вимогам:

- файл у форматі .xlsm, .xlsx;
- містить вкладка з назвою «Students»;
- всі ячейки таблиці заповнені.

Таблиця на вкладці «Students» має вигляд:

- головка. Рядок №1, містить назви стовпців, що мають бути заповнені для здійснення масової реєстрації;
- тіло. Рядки починаючи з 2, стовпці А-С (табл. 2.1). Містять дані зібрані, перевірені та підготовлені старостою для передачі викладачу.

Таблиця 2.1 – Опис стовпців таблиці для масової реєстрації

Назва стовпця	Призначення	Формат	Обов'язковий
Group	Автоматичне включення слухача до групи в системі	Код групи	Так
FullName	Ідентифікація користувача викладачем для виставлення оцінки, використання для звернень до користувача з боку системи	Текст (без обмежень визначеним шаблоном)	Так

Назва стовпця	Призначення	Формат	Обов'язковий
Email	На вказану адресу буде вислано лист з даними для входу, або інструкціями щодо подальшої реєстрації	Email	Так

Імпорт файлу здійснюється користувачем у ролі «Викладач» з панелі керування користувачами. Після успішного завершення процесу, система має повідомити користувача, шляхом показу текстового повідомлення, що імпорт завершено успішно.

Інші вимоги до масової реєстрації:

- якщо адреса електронної пошти користувача вже зареєстрована в системі імпорт не припиняється, даний користувач пропускається оскільки він уже має обліковий запис, оновлення даних не відбувається, група не змінюється;
- код групи може бути написаний в будь-якому регістрі, коди «ІТ-83МП», «іт-83мп», «Іт-83мп» тощо – вважаються однаковими та ідентифікують одну групу;
- якщо під час імпорту зустрічається код групи, який не існує в системі – створюється нова група з цим кодом;
- можливий імпорт студентів різних груп одночасно, кожен з них має бути віднесений до вказаної групи;
- студент не може бути в кількох групах одночасно;
- шаблон електронного файлу формату .xlsx викладач може завантажити з панелі керування користувачами;
- у випадку внесення змін до системи існуючі облікові записи повинні бути розширені без видалення чи зміни унікального ідентифікатора.

Нефункціональна вимога до процесу реєстрації (рис. 8) через викладача одна – масова реєстрація через імпорт будь-якої кількості студентів повинна потребувати не більше ніж 4 дії в системі:

- завантаження шаблону;
- перехід на вкладку «Слухачі»;
- завантаження заповненого файлу;
- імпорт файлу до системи.

У випадку, якщо деякі дані не заповнено, або реєстрація облікового запису неможлива викладачу має бути показане повідомлення з попередженням про це.

Якщо обліковий запис із вказаною у файлі адресою електронної пошти вже зареєстровано в системі на цю адресу має надійти повідомлення про спробу надання доступу до платформи. Ніякі дані (логін, пароль) у такому випадку не генеруються та не надсилаються.

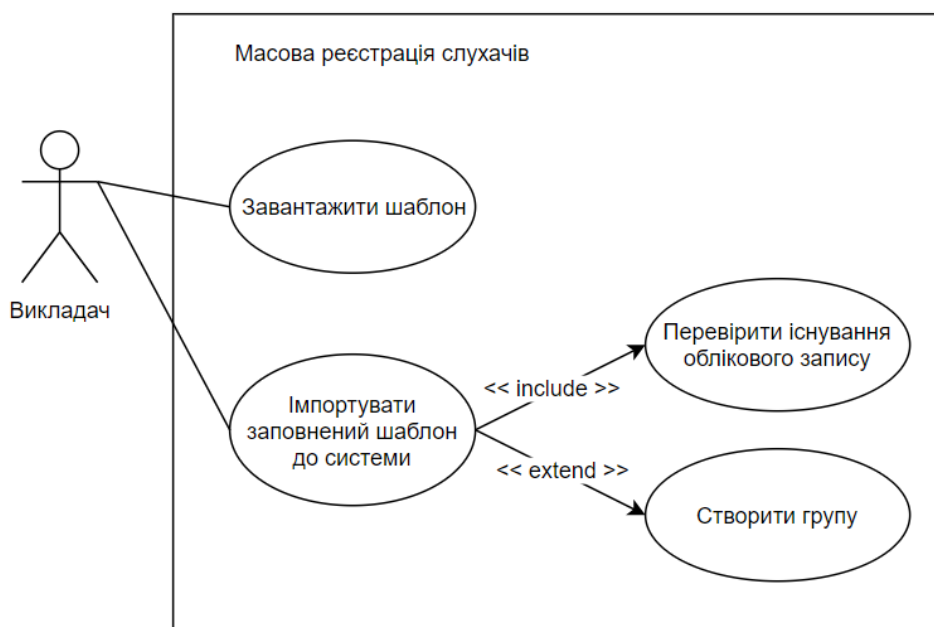


Рисунок 8 – Діаграма прецедентів масової реєстрації слухачів

Функціональність створення, зміни, видалення, відкриття для слухачів і запуску курсу доступна користувачу в ролі «Викладач», причому для редагування та видалення доступні тільки курси створені активним користувачем.

Створення курсу можливе такими способами:

- створення курсу з панелі керування курсами;
- імпорт стороннього курсу.

Панель керування курсами відображає всі доступні користувачу курси з їх статусом (активний, неактивний), назвою та коротким описом, а також признаком чи є курс імпортованим, у алфавітному порядку за назвою.

Створення курсу відбувається шляхом внесення до системи наступних даних:

- назва курсу. Унікальність не вимагається, проте рекомендована;
- опис. Короткі відомості про курс: мета, ідея тощо;
- кредити. Кількість кредитів, для формування кінцевого звіту;
- години. Кількість годин, для формування кінцевого звіту;

Курс стає доступний слухачам тільки після його запуску викладачем.

Кожен курс поділяється на теми. Тема в контексті системи – відокремлена частина курсу з власною назвою, описом та наповненням.

Після створення курсу (описано вище) викладач має наповнити його темами. Процес наповнення виглядає аналогічно до створення курсу і полягає у внесенні до системи наступних даних:

- назва теми. Унікальність не вимагається, проте рекомендована;
- опис. Короткий опис теми, рекомендовано вказувати основну ідею теми;
- статичний бал. Максимальна кількість рейтингових балів за проходження теми;
- номер. Порядковий номер теми (проставляється автоматично).

Порядок наповнення тем не важливий, щоб приступити до наповнення потрібно перейти до теми та заповнити запропоновані пункти. В межах системи тема може бути наповнена такими типами внесених даних:

- теоретичні відомості. Заплановано, що це буде конспект лекцій в електронній формі доступний прямо зі сторінки системи при відкритті теми;
- документ. Підтримується формат .pdf, як найпоширеніший документний та той що зберігає форматування незалежно від того де відкривається;
- посилання. Іменоване посилання на сторонній ресурс;
- тест. Оцінка знань із пройденої теми в тестовій формі.

Всі типи повинні бути прив'язуються до теми, мають признак обов'язковості для успішного завершення теми та відповідають єдиному інтерфейсу, проте можуть бути розширеними (рис. 9).

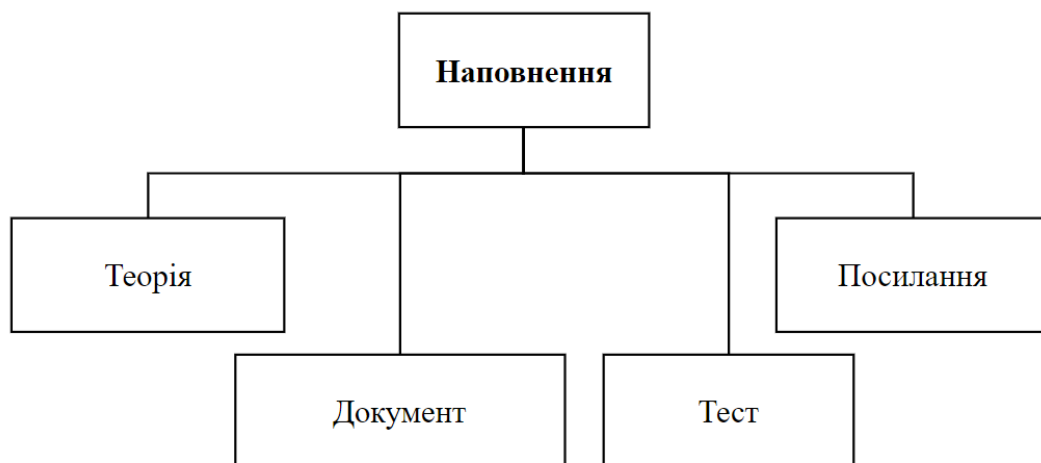


Рисунок 9 – Різновиди типів даних для наповнення теми

Дані які є загальними для всіх типів наповнення, та повинні використовуватися при оцінюванні, визначенні що за тим представлено, та до якої теми належить описано в таблиці 2.2.

Таблиця 2.2 – Опис загальних членів для всіх типів наповнення

Назва члена	Тип	Призначення	Значення по замовчуванню
Батько	Код	Ідентифікує тему до якої призначено наповнення	Немає, обов'язковий член
Тип	Теорія Посилання Документ Тест	Ідентифікація типу наповнення	Залежно від типу наповнення, вказується автоматично
Чи обов'язкове	Так чи Ні	Дія з наповненням обов'язкова для завершення теми	Ні
Відсоток від загальної оцінки	Кількість балів	Відсоток від вказаної кількості балів за тему, нараховується за дію з наповненням	0

Крім наведених членів кожен тип наповнення містить власні властивості. Опис додаткових членів типу «Теорія» наведено в таблиці 2.3.

Таблиця 2.3 – Додаткові члени типу «Теорія»

Назва члену	Тип	Призначення	Значення по замовчуванню
Текст	Текст без розмітки	Текст для режиму читання	Немає
Стилізований текст	Розмітка веб сторінки	Розмітка теоретичних відомостей для відображення на веб-сторінці	Немає

Відповідно до вимог в тему може бути завантажено документи у форматі .pdf. Ця вимога виконується через реалізацію типу «Документ», члени якого описано в таблиці 2.4.

Таблиця 2.4 – Опис додаткових членів наповнення типу «Документ»

Назва члену	Тип	Призначення	Значення по замовчуванню
Назва файлу	Рядок	Назва файлу для відображення	Немає
Розмір	Число з 2 знаками після коми	Розмір файлу в МБ	Немає
Код файлу	Код	Унікальний ідентифікатор файлу в системі	Немає

Систематизація додаткових ресурсів, файлів, що не можуть бути завантажені до системи відбувається через тип «Посилання», згідно вимог – це іменоване посилання на деякий ресурс. Додаткові члени описано в таблиці 2.5.

Таблиця 2.5 – Опис додаткових членів наповнення типу «Посилання»

Назва члену	Тип	Призначення	Значення по замовчуванню
Назва	Рядок	Назва посилання для відображення	Url

Назва члену	Тип	Призначення	Значення по замовчуванню
Посилання	Рядок	Посилання	Немає

Всі типи наповнення можуть бути відредаговані користувачем, який їх створив. Ідентифікація користувача не відокремлюється від ідентифікації власника курсу, оскільки наповнення курсу можливе тільки через обліковий запис користувача, який створив цей курс.

Якщо користувачу доступне редагування або наповнення курсу він автоматично має доступ до наповнення тем та редагування описаних раніше типів.

Крім наведених додаткових членів всі типи наповнення містять технічні поля, що заповнюються автоматично. Дані, про які йдеться описано в таблиці 2.6. Їх передбачено з метою спрощення логування та відловлення помилок. Окрім цього, у випадку необхідності їх можна використати для додаткових функцій системи. Сюди можна включити групування та сортування за датою створення.

Таблиця 2.6 – Технічні члени присутні у всіх типів наповнення

Назва члену	Тип	Призначення	Значення по замовчуванню
Створено ким	Текст	Логін користувача який створив тип	Own
Дата створення	Дата і час	Дата створення типу	Поточна дата

Тест відокремлюється від інших типів наповнення, оскільки є набагато складнішим за своїм призначенням, та повинен відповідати наступним вимогам:

- одна тема може містити не більше одного тесту;
- тест є необов'язковим елементом;
- база питань для тесту може містити більше питань, ніж потрібно для проведення оцінювання, для слухача питання вибираються випадковим чином, а їх кількість задається при створенні тесту;
- слухач може проходити тест необмежену кількість раз – актуальною вважається оцінка отримана за останню спробу;

- всі спроби користувача реєструються в системі, та використовуються для відображення статистики на панелі викладача;
- для уникнення шахраювання з боку студентів методом скріншотів правильних відповідей слухач не може бачити правильну відповідь, також він не знає чи була дана ним відповідь неправильною. Результат оцінювання – відсоток правильних відповідей;
- викладач має доступ до відповідей слухачів;
- за тест можуть нараховуватися додаткові бали (понад максимальну кількість вказану для теми).

Тест – складна сутність, яку можна описати діаграмою зв'язків зображеною на рисунку 10. Тут наочно зображено вимоги:

- не більше 1 тесту на тему;
- тест має багато завдань, але завдання не існує без тесту;
- на кожне завдання є багато варіантів відповідей, кожен з варіантів прив'язаний до завдання.

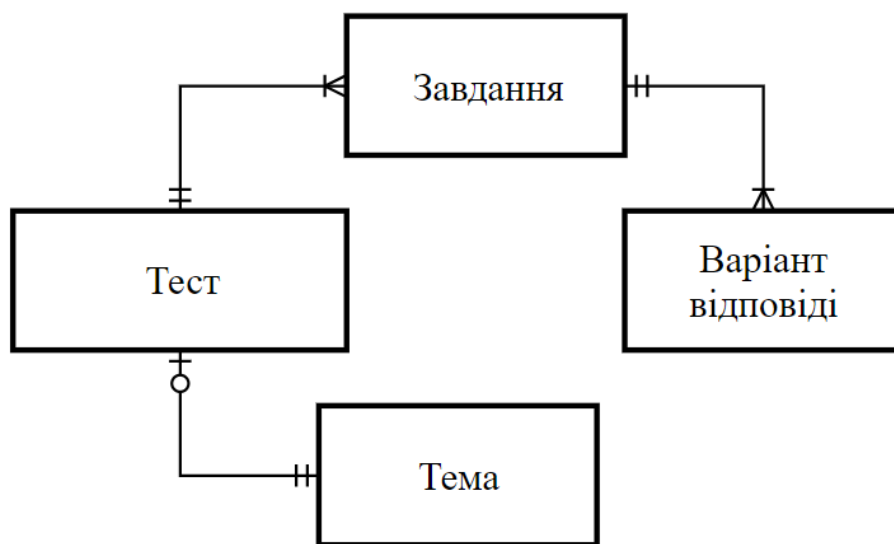


Рисунок 10 – Діаграма-опис для тесту

Крім описаних вище зв'язків тест містить наведені раніше технічні члени та дані, які описано в таблиці 2.7.

Таблиця 2.7 – Опис додаткових даних типу наповнення «Тест»

Назва члену	Тип	Призначення	Значення по замовчуванню
Бонусні бали	Ціле число	Кількість бонусних балів за проходження тесту на 100%	0
Необов'язковий опис	Текст	Необов'язковий короткий опис тесту	Немає
Бібліотека завдань	Список завдань	Список доступних для тесту питань, забезпечує зв'язок 1 до багатьох між тестом та завданнями	Немає
Кількість запитань для студента	Число	Кількість питань	Немає

Завдання тесту не є наповненням, тому набір членів у ньому відрізняється, крім технічних. Дані кожного завдання однакові, їх наведено в таблиці 2.8.

Важливо відмітити, що перша версія системи не підтримує завдань, крім текстових, але передбачено можливість розширення через інтеграцію зображень, розгорнутих відповідей тощо.

Таблиця 2.8 – Опис членів типу Завдання тесту

Назва члену	Тип	Призначення	Значення по замовчуванню
Код завдання	Код	Унікальний ідентифікатор завдання в межах системи	Немає
Текст	Текст	Обов'язковий член, містить текст запитання	Немає
Підказка	Текст	Текстова підказка для слухача	Немає
Код тесту до якого належить	Код або назва	Ідентифікатор тесту в який питання додається	Немає
Варіанти відповідей	Список відповідей	Список варіантів відповідей для питання	Немає

У перших версіях системи підтримуються лише текстові завдання з текстовими відповідями, проте, як сказано раніше, заплановано введення інших типів. Дані варіантів відповідей описано в таблиці 2.9.

Таблиця 2.9 – Опис даних варіанту відповіді на питання тесту

Назва члену	Тип	Призначення	Значення по замовчуванню
Ідентифікатор	Код	Унікальний ідентифікатор відповіді в межах системи	Немає
Текст варіанту відповіді	Текст	Обов'язковий член, містить текст відповіді	Немає
Чи правильний	Так чи Ні	Признак правильної відповіді	False
Код завдання	Код	Ідентифікатор завдання в яке варіант відповіді додається	Немає

Окрім створення курсу з системи, має бути можливість інтеграції сторонніх курсів, що забезпечують зі свого боку реалізацію API для імпорту курсу та результатів.

Сценарій імпорту та інтеграції стороннього курсу виглядає так:

- викладач отримує посилання для імпорту на сторонньому ресурсі, копіює його у поле вікна імпорту курсу;
- система робить GET запит за вказаною адресою та отримує відповідь у форматі JSON з даними, структура яких описана в таблиці 2.10;
- на основі отриманих даних у системі створюється власний курс, відкриття якого переправляє слухача на ресурс, з якого було зроблено імпорт, відкриття курсу супроводжується передачею сторонньому сервісу ідентифікатора облікового запису для майбутнього експорту результатів;
- для отримання результатів після відповідної дії викладача (імпорт результатів) система робить GET запит за адресою імпорту та параметром «результати» на що отримує відповідь у форматі JSON;
- відповідь оброблюється системою та зберігається у формі, яка забезпечить використання даних для генерації статистики слухача.

Таблиця 2.10 – Структура відповіді для успішного імпорту курсу

Назва члена	Тип	Призначення	Обов'язковий	Значення по замовчуванню
Title	Текст	Назва курсу	Так	Немає
ExternalId	Код або Назва	Ідентифікатор курсу у сторонній системі	Так	Немає
OpenUrl	Рядок	Посилання на курс для слухача	Так	Немає
Credits	Ціле число	Кількість кредитів	Ні	4
Hrs	Ціле число	Кількість годин	Ні	192
ChaptersMetadata	Список тем	Метадані тем курсу	Ні	Пуста колекція

Всі члени колекції ChapterMetadata для сумісності із темами курсів у системі повинні відповідати структурі наведеній у таблиці 2.11.

Таблиця 2.11 – Структура метаданих тем

Назва члена	Тип	Призначення	Обов'язковий	Значення по замовчуванню
Title	Текст	Назва теми	Так	Немає
ExternalId	Код	Ідентифікатор теми у сторонній системі	Ні	Немає
PossiblePoints	Ціле число	Максимальна кількість балів за тему	Ні	<u>100</u> кількість тем
Description	Текст	Опис теми	Ні	Немає

Сценарії використання функціоналу імпорту курсу зображені на рисунку 11.

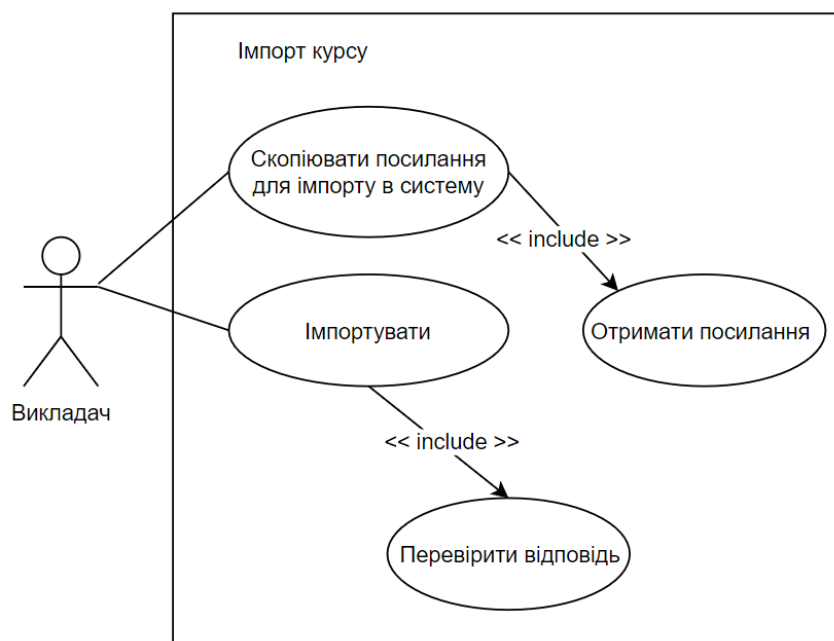


Рисунок 11 – Діаграма прецедентів імпорту стороннього курсу

Користувач у ролі «Викладача» може слідкувати за успіхами слухачів, у межах створених ним курсів. Панель контролю забезпечує різнопланову функціональність моніторингу:

- журнал оцінок;
- активність групи;
- активність користувача;
- кращі результати;
- список студентів без активності;
- детальний аналіз активності вибраного студента.

З панелі контролю викладач може здійснити експорт у .xlsx файл кінцевих оцінок слухачів. У таблиці 2.12 описано структуру результуючого файлу.

Таблиця 2.12 – Структура файлу з результатами

Назва стовпця	Призначення
FullName	Повне ім'я слухача
Course	Назва курсу
TheoryPoints	Бали за теорію
TestPoints	Бали за тести
BonusPoints	Бонусні бали

Назва стовпця	Призначення
Result	Кінцевий результат

Система повинна зберігати такі дії слухача з курсом:

- відкриття теми;
- перехід за посиланнями;
- прогорткування теорії;
- відкриття документів;
- кожену спробу проходження тесту з її результатом;
- кожену дану відповідь на питання тесту;
- перехід до стороннього курсу.

На основі активності слухачів будується статистика групи до якої ці слухачі відносяться. Ключовим показником для групи є середній бал учасників (крім тих, що не мають жодної спроби). Окремо виділяються слухачі які пройшли курс з максимальним балом. Та студенти від яких немає активності в курсі.

Детальний аналіз вибраного слухача відбувається при відкритті панелі моніторингу, тут викладач може побачити:

- кількість спроб проходження тесту;
- середній відсоток правильних відповідей;
- кінцевий результат;
- список питань на які дано неправильні відповіді та які відповіді дано;
- список виправлених питань;
- кількість відвідувань теми та бали набрані за теорію;
- кількість відкриттів документів;
- додаткові бали нараховані за 100% правильно виконаний тест;
- загальний бал.

Нефункціональні вимоги, що покриваються наведеними функціональними:

- інтеграція до навчального закладу;
- оцінка поставлена системою не є кінцевою, викладач але викладач може врахувати її проаналізувавши активність слухачів;

- неможливість шахраювання при проходженні тестів.

На основі поставлених вимог можна прослідкувати орієнтацію системи на викладача, демонстрація роботи повинна проводитися шляхом забезпечення курсом для предмету комп'ютерної криптографії на еліптичних кривих. Для можливості безкоштовного впровадження система забезпечується обліковим записом викладача для ознайомлення. Конфігурація системи відбувається шляхом заповнення даних у файлі `appSettings.json` у секції `TeacherAccounts` яка є об'єктом, з такими членами:

- `System`. Масив облікових записів, які включаються до системи по замовчуванню та мають виключні права.
- `Additional`. Масив облікових записів, які додаються в систему з часом.

Після перезапуску системи дані з цієї секції підтягуються в базу даних системи, щоб, таким чином, надати доступ через нові облікові записи.

3. РОЗРОБЛЕННЯ СЦЕНАРІЇВ ВИКОРИСТАННЯ СИСТЕМИ

3.1 Актори та залежності сценаріїв використання

Перед тим, як описувати сценарії використання системи важливо визначити типи користувачів, які можуть отримати доступ до системи. Відповідно до вимог користувачі повинні поділятися на викладачів та слухачів, для них у системі буде виділено відповідні ролі. Залежно від ролі відрізняється функціональність, яку забезпечує система. Крім того виділено актора «Гість» - неавторизований користувач.

В таблиці 3.1 описано всі можливі для гостя варіанти використання системи.

Таблиця 3.1 – Варіанти використання для актора «Гість»

Варіант використання	Опис
Авторизуватися	Система надає користувачу форму для введення логіну та пароля. Після здійснення авторизації Гість отримує роль «Викладач» або «Студент» в залежності від внесених у систему даних.

Особливості ролі «Викладач» відповідно до вимог:

- немає прямої реєстрації, облікові записи створюються через файл конфігурації;
- виконує функцію адміністратора та наповнювача;
- має доступ до панелі контролю результатів та активності слухачів.

Незважаючи на короткий перелік особливостей ролі саме викладач має найбільше можливих варіантів використання.

Опис доступних викладачу варіантів використання системи поділено на 2 частини (окремі таблиці):

- адміністрування слухачів;
- адміністрування курсів.

В таблиці 3.2 наведено варіанти використання, що може здійснювати «Викладач» для адміністрування слухачів.

Таблиця 3.2 – Прецеденти доступні ролі «Викладач» для керування слухачами

Варіант використання	Опис
Керувати користувачами	Система надає панель керування користувачами, містить функціональність створення групи, реєстрації користувача, масової реєстрації користувачів, перегляду списку існуючих користувачів, зміни групи користувачу
Створити групу	Створення групи відбувається з панелі керування користувачами, шляхом заповнення коду групи у відповідній формі біля переліку існуючих груп
Зареєструвати користувача	Система забезпечує викладача формою для вводу даних нового користувача – повного імені, адреси електронної пошти, коду групи. У випадку існування облікового запису система показує відповідне повідомлення
Імпортувати користувачів	Система забезпечує викладача можливістю завантаження файлу у форматі .xlsx або .xlsm, що заповнений у відповідному до вимог форматі та здійснити масове створення облікових записів із автоматичною прив'язкою користувача до групи через її код
Переглянути групи	Система відображає таблицю з існуючими групами в алфавітному порядку їх кодів, та кількість користувачів у групі
Переглянути список користувачів	Система відображає таблицю з існуючими обліковими записами, розташованими за алфавітним порядком коду групи та повного імені користувача, тут також виводять адреси електронної пошти
Подивитися відповіді слухачів	Система не дає студенту можливості побачити правильну відповідь до завдання тесту, але завдяки інтеграції у навчальний процес у вибраний час викладач може провести очну роботу над помилками, після чого студенти можуть заново пройти тест і покращити результати
Оцінити студента	Система не здійснює остаточного та кінцевого оцінювання, панель моніторингу виступає інструментом для здійснення оцінювання та повинна наочно показувати активність та спроби з усіма результатами слухачів

Окрім можливості керування доступами до системи та даними слухачів, користувач у ролі «Викладач» відповідає за адміністрування курсів. Варіанти використання для цього виділено в таблиці 3.3.

Таблиця 3.3 – Прецеденти доступні ролі «Викладач» для керування курсами

Варіант використання	Опис
Адмініструвати курси	Панель адміністрування курсів забезпечує викладача функціями створення курсів з системи, редагування метаданих, наповнення та відкриття доступу до курсу
Створити курс	Система забезпечує кілька методів створення курсу. Створення з панелі адміністрування курсів – система надає форму для заповнення назви, опису, кількості кредитів, та кількості годин. Імпорт стороннього курсу – система надає форму для заповнення наданого стороннім сервісом шляху імпорту.
Імпортувати курс	Система забезпечує автоматичний імпорт даних зі стороннього сервісу, що реалізує API експорту у валідному форматі. При такому варіанті використання наповнення курсу непотрібне, курс вважається створеним та готовим до запуску
Наповнити курс	Система забезпечує викладача функціональністю створення тем, що означає внесення до системи назви, опису, максимальної кількості балів за тему. Та можливість наповнення, відповідно до вимог, теорією, .pdf документами, посиланнями, створення тесту
Створити тест	Система забезпечує викладача панеллю створення тесту, що відбувається шляхом введення у систему опису тесту, списку питань із варіантами відповідей та вказівником на правильну
Відкрити курс	Доступ до курсу надається шляхом вказування групи, що має доступ до курсу, або конкретних користувачів шляхом введення коду групи у форму, або повного імені користувача, відповідно
Запустити курс	Система надає одноіменну кнопку для запуску курсу, що робить його доступним для користувачів, яким курс відкрито
Отримати результати	Система забезпечує викладача панеллю моніторингу результатів і активності слухачів. З якої шляхом відкриття запропонованих системою посилань є можливість стежити за історією дій слухачів,

Варіант використання	Опис
	ознайомитися зі загальними результатами, успішністю групи в цілому, та детально подивитися активність вибраного слухача
Імпортувати результати	Система забезпечує викладача кнопкою, що дозволить отримати результати слухачів із сторонніх сервісів для раніше імпортованих курсів, якщо реалізовано відповідне API
Експортувати оцінки	Система забезпечує функцію експорту балів у .xlsx файл

Роль слухача в системі називається «Студент», можливості що система забезпечує для такого користувача:

- реєстрація через лист на електронну пошту;
- реєстрація через викладача;
- масова реєстрація аккаунтів через імпорт;
- перегляд доступних курсів;
- робота з запропонованими матеріалами;
- тестування знань.

Опис варіантів використання що доступні ролі «Студент» наведено та описано в таблиці 3.3.

Таблиця 3.3 – Прецеденти доступні користувачу в ролі «Студент»

Варіант використання	Опис дії варіанта використання
Переглянути список курсів	Система відображає список відкритих для студента активних курсів, у випадку повного проходження відображається кінцева оцінка за курс
Відкрити курс	Система надає користувачу деталі курсу, назву, опис та список тем, які входять у курс
Відкрити тему	Система записує відкриття теми для панелі моніторингу, відкриває тему та показує наповнення
Відкрити посилання	Відкриття стороннього ресурсу відбувається шляхом натиснення на його назву, система відкриває

Варіант використання	Опис дії варіанта використання
	посилання у новій вкладці, система реєструє відкриття для моніторингу
Відкрити документ	Відкриття .pdf файлу відбувається шляхом надання доступу до статичного ресурсу в новій вкладці, відкриття файлів реєструється для моніторингу
Читати теорію	Теорія може бути відкрита в режимі читання, прогорткування теорії реєструється для моніторингу викладачем
Пройти тест	Система дає можливість пройти тест студент, який закінчив читання теорії. Кількість спроб необмежена, актуальний останній результат. Кожна спроба реєструється системою для моніторингу. Результат проходження відображається у відсотковій формі, можливості переглянути правильність даної відповіді немає. Робота над помилками проводиться через панель моніторингу результатів із облікового запису викладача

Загальну діаграму прецедентів представлено в додатку А а детальний опис кожного з варіантів використання в розділі 3.2.

3.2 Опис сценаріїв використання

В даному розділі зроблено акцент на детальному описі сценаріїв використання:

- панелі керування користувачами;
- панелі адміністрування курсів;
- панелі моніторингу результатів;
- реєстрації активності слухача.

Після входу в систему та авторизації викладач переправляється на головну сторінку системи, звідки може відкрити панель керування користувачами. Відкриття здійснюється натисненням на одноіменне посилання. Відкриття панелі відбувається

шляхом завантаження відповідної сторінки. Відкриття панелі вважається успішним якщо:

- сторінку завантажено;
- стилі та скрипти відпрацьовують без помилок;
- сторінка відображає таблицю зі списком існуючих груп;
- сторінка відображає таблицю зі списком зареєстрованих користувачів;
- під час введення даних у поля пошуку відбувається фільтрація по таблицях;
- натиснення на кнопку «+» біля назви таблиці відкриває панель створення;
- натиснення на кнопку «Імпорт» відкриває діалогове вікно вибору файлу.

Таблиця зі списком існуючих груп містить наступні стовпці:

- код групи;
- кількість слухачів;
- дата створення.

Таблиця зі списком користувачів містить такі стовпці:

- код групи;
- повне ім'я;
- адреса електронної пошти;
- дата реєстрації.

Для створення групи після натиснення кнопки «+» потрібно заповнити такі дані:

- код групи.

Для реєстрації користувача після натиснення кнопки «+» потрібно внести до системи такі дані (всі поля обов'язкові):

- код групи;
- повне ім'я;
- адреса електронної пошти.

Для здійснення імпорту користувачів потрібно:

- натиснути кнопку «Імпорт»;
- вибрати заповнений файл формату .xlsx або .xlsm;

- дочекатися завантаження файлу та повідомлення про успішне завершення імпорту.

Редагування облікового запису користувача відбувається шляхом зміни даних у формі, що відображається після натиснення на кнопку «Редагувати» в рядку слухача.

Дані, що можуть бути змінені:

- код групи;
- повне ім'я.

Генерація паролю для входу відбувається автоматично, дані для входу в систему відправляються на вказану адресу електронної пошти та вважаються правильними до моменту внесення у них змін.

Відкриття панелі керування курсами відбувається шляхом натиснення на одноіменне посилання. Сторінка вважається завантаженою, якщо:

- відображаються курси які доступні для адміністрування викладачу, або повідомлення що доступних курсів немає;
- кнопка «+» відкриває форму для створення курсу;
- кнопка «Імпорт» відкриває форму для вставки ресурсу для імпорту;
- натиснення на курс відображає теми або повідомлення, що тем ще немає.

Для створення курсу потрібно внести до системи такі дані:

- назва курсу;
- опис курсу;
- кількість кредитів;
- кількість годин.

Після натиснення на кнопку «Створити» та успішного створення курсу в системі відкривається список тем (сторінка адміністрування тем). Всі створені з цієї сторінки теми будуть додані до відкритого курсу, назва якого відображується на головці панелі. Для створення теми в систему мають бути внесені такі дані:

- назва;
- опис;

- максимальна кількість балів за завершення теми (без врахування бонусних балів).

Після створення теми не відбувається автоматичний перехід до сторінки її наповнення. Для того щоб перейти на неї потрібно натиснути «Наповнення» на панелі теми.

На сторінці наповнення відокремлено відображаються такі форми:

- теорія, для заповнення рекомендовано використовувати вбудований редактор, проте підтримується також користувацька HTML розмітка;
- файли, можливе завантаження . pdf файлів (не більше 5 за раз);
- посилання, форма для динамічного додавання посилань, потрібно заповнити поля ім'я та Url;
- тест, кнопка «Створити тест».

Після успішного наповнення курсу контентом він може бути опублікований (запущений). Публікація курсу – процес підтвердження коректності даних (заповнення всіх обов'язкових форм) та відкриття доступу для вибраних слухачів. Вибір слухачів відбувається з панелі курсу одноіменною кнопкою «Відкрити» натиснення на яку відкриває форму з необов'язковими полями для вводу номеру групи, імені студента (випадаючий список з автозаповненням) або електронної адреси слухача.

На панелі курсу відображається посилання на сторінку результатів. Ця сторінка збирає та забезпечує наступні прецеденти:

- отримати результати;
- переглянути відповіді на тести;
- імпортувати результати стороннього курсу;
- оцінити студента;
- експортувати оцінки.

Прецедент «Отримати результати» виникає автоматично під час завантаження сторінки. Суть його відображується в отриманні, обробці та відображенні даних про активність слухачів курсу в наступних формах:

- загальна активність слухачів, середня активність;

- активність груп;
- найкращі результати.

Перегляд відповіді на тести відбувається за рахунок натиснення на відповідно кнопку поряд з прізвищем слухача. Система відображає всі спроби, проте завантажує дані тільки для останньої. Завантаження даних для попередніх спроб можливе після натиснення на відповідну кнопку поряд з номером спроби. Окрім відповіді слухача відображається також правильна відповідь (якщо така є).

Прецедент «Імпортувати результати» є прямою передумовою для оцінювання слухача і полягає в натисненні одноіменної кнопки, реакцією на що є виконання запиту до стороннього курсу з відповіддю на який система отримує результати проходження курсів у партнерів і потенційну рекомендовану оцінку. Якщо дані вже існують в системі а імпорт зроблено повторно система перезаписує існуючі дані.

«Експортувати оцінки» - прецедент сформовано для автоматизації рутинної роботи викладача відображається одноіменною кнопкою, реакцією на натиснення якої є формування Excel документа наступної структури:

- група;
- повне ім'я слухача;
- загальний бал за теорію;
- загальний бал за опрацювання додаткових джерел;
- загальний бал за тести;
- загальна кількість додаткових балів;
- кінцева оцінка.

Після успішного формування відкривається вікно вибору теки для зберігання документа.

Описані форми та функції доступні користувачу в ролі «Викладач», щодо студентів та користувачів у ролі «Слухач» перелік можливих прецедентів суттєво обмежений в зв'язку з неможливістю передачі суттєвого контролю над можливостями системи студентам та слухачам.

На головній сторінці панелі слухача відображаються панелі доступних курсів з назвою, контактами викладача та загальною кількістю тем. Знизу панелі відображається відсоток виконання курсу.

Прецедент «Відкрити курс» виникає шляхом натискання на панелі курсу, реакцією на це є відкриття сторінки курсу на якій відображаються такі дані:

- назва курсу;
- короткий опис;
- контакти викладача;
- список тем у формі панелей з можливістю вибору;
- прогрес у проходженні теми та кількість балів за неї.

Прецедент «Відкрити тему» виникає внаслідок натиснення на панелі теми на сторінці курсу.

Реакцією системи на цю подію є відкриття сторінки теми включно з завантаженням теоретичних матеріалів, посилань, документів та даних про тест.

На цій сторінці можливі наступні прецеденти:

- відкрити документ;
- відкрити посилання;
- розпочати тестування.

Прецедент «Читати теорію» виникає автоматично при завантаженні даної сторінки, при натисненні на кнопку «Показати повністю». Система реєструє всі дії слухачів і зберігає протягом усього курсу. Окрім наведених прецедентів записуються також наступні події:

- покидання вкладки;
- перехід на іншу сторінку.

Всі ці прецеденти дозволяють системі забезпечити користувачів тими функціями, що визначені при формуванні вимог і відповідають їм.

Даний опис дозволяє розпочати аналіз ринку існуючих технологій та обрати ті, що найкраще відповідають вимогам системи та кваліфікації розробника.

4. ВИБІР ТА ОБГРУНТУВАННЯ ЕЛЕМЕНТІВ ТА ТЕХНОЛОГІЙ

4.1 Огляд варіантів реалізації

Ключовою нефункціональною вимогою системи є відсутність необхідності встановлювати програмне забезпечення на машині слухача. Критичні вразливості настільних застосунків через які варто обрати іншу модель програмного комплексу наступні:

- незахищеність від декомпіляції, підміни значень змінних;
- залежність від платформи, необхідність підтримки декілької рішень;
- небажання слухачів інсталювати стороннє програмне забезпечення;
- рівень технічної забезпеченості викладачів у закладах освіти.

Іншими варіантами є розробка мобільних застосунків або веб-додатку. Через поширеність платформ iOS та Android знову постає питання розробки більше ніж 1 версії застосунків [9]. Через це, та зважаючи на фактор необхідного невторчання у вихідний код доцільно вибрати клієнт-серверну модель системи. Вибравши цю альтернативу отримано наступні позитивні фактори:

- логіка оцінювання, зберігання даних та результатів відділена від користувача та виконується на сервері, що підвищує рівень захищеності;
- клієнт-серверна модель дозволяє вже на початковому етапі розробки враховувати можливість інтеграції сторонніх електронних навчальних курсів;
- для використання системи (викладачу і слухачу) достатньо браузера та доступу до мережі Інтернет;
- веб-застосування може бути відкрите мобільним браузером.

4.2 Вибір бази даних та сховища

Враховуючи вимоги до системи необхідно вибрати сховище даних та файлів.

На основі кваліфікації розробника та популярності технологій до вибору представлено такі альтернативи:

- нереляційна БД MongoDB;
- документна база даних реального часу Google Firebase;
- СУБД MS SQL Server.

Важливо звернути увагу, що на ринку представлено багато інших варіантів, проте вивчення документації та реалізація проекту з використанням цих технологій, імовірно, займе забагато часу та може викликати несподівані помилки.

Для сховища завантажених файлів до вибору пропонуються наступні варіанти:

- файлове сховище сервера на якому буде працювати веб-рішення;
- Azure Blob;
- Amazon S3.

Відповідно до опису технологій необхідно проаналізувати кожен з наведених альтернатив і обрати ту, що якнайповніше задовольнить потреби сформовані за рахунок вимог.

MongoDb - це база документна база даних, яка зберігає дані в JSON-подібних документах. Розробники вважають, що це найприродніший спосіб думати про дані і набагато виразніший та потужніший, ніж традиційна реляційна [10].

Ключовими перевагами MongoDB є:

- JSON структура документів;
- потужна та швидка мова запитів;
- 2 типи зв'язки;
- безкоштовна для обмеженої кількості запитів;
- горизонтальна розширювальність;
- динамічна схема моделі даних.

Щодо недоліків варто зазначити наступні:

- відсутність підтримки транзакцій;
- немає підтримки тригерів;
- конкурентний доступ відсутній.

Подібною структурою та перевагами відзначається також Google Firebase, проте концепція цього сховища дещо відрізняється.

Google Firebase - це мобільна платформа Google, яка допомагає швидко розвивати високоякісні додатки та розвивати свій бізнес. Вже за визначенням очевидно, що інструмент забезпечує користувача не тільки повноцінним сховищем даних а й додатковими можливостями. Firebase працює на основі JSON [11].

Переваги даної платформи наступні:

- швидка розробка клієнт-серверних застосунків;
- документна БД;
- вбудовані засоби аналітики, обміну повідомленнями;
- real time оновлення;
- дозволяє будувати серверну логіку;
- вбудована аутентифікація;
- сховище файлів.

Недоліки платформи:

- залежність від провайдера;
- ціна;
- враховуючи вбудовану підтримку real time БД доступна безкоштовно кількість запитів є недостатньою для системи середньої складності;
- всі дані рішення передаються Google для опрацювання та аналітики.

Щодо СУБД MS SQL Server яка відрізняється використанням T-SQL для запитів і є пропріетарною технологією Microsoft. Для розробника доступна версія Express яка має обмежену функціональність, проте забезпечує доступ до всіх потрібних можливостей СУБД. До переваг варто віднести:

- підтримка від компанії зі світовим іменем;
- спрощені розгортання, передача та інтеграція великий баз даних;
- реляційна модель зберігання даних;
- висока швидкодія;
- доступність транзакцій;
- T-SQL;

- індексування (в т.ч. для повнотекстового пошуку).

Недоліки:

- ПЗ пропріетарне за БД розміщену в хостинг провайдера доведеться додатково платити;
- ціна (без доступних програм від Microsoft);
- необхідна кваліфікація для написання ефективних запитів на T-SQL [12].

Основні критерії для вибору сховища даних на основі сформованих вимог:

- підтримка транзакцій;
- доступність;
- можливість резервного копіювання;
- швидкодія.

Аналіз відповідності альтернативи обраним критеріям наведено в таблиці 4.1.

Таблиця 4.1 – Аналіз відповідності альтернативи критеріям вибору

Критерій	MongoDb	Firebase	MS SQL Server
Підтримка транзакцій	Немає	Немає	Є
Ціна (\$/год)	0.02	0.04	0.007
Резервне копіювання	Додатково	Автоматично	Налаштовується
Швидкодія	Безперебійна навіть із додатками із високим трафіком	Поступається MongoDB	Поступається MongoDB в запитах з великою кількістю з'єднань

Враховуючи наведені характеристики варто відзначити, що незважаючи на високу швидкодію MongoDB і присутність можливості резервного копіювання бази даних MSSQL Server має вбудовану підтримку транзакцій, що може значно вплинути на якісь таких функцій системи:

- імпорт даних з Excel файлу;
- імпорт стороннього курсу;
- експорт оцінок;
- імпорт результатів і оновлення даних.

Враховуючи також присутність на ринку хостинг провайдерів у вартість послуг яких входить не тільки розміщення бази даних а й хостинг веб-застосування вибір MS SQL буде доцільним.

Враховуючи це та вимоги до системи необхідно визначити критерії вибору сховища файлів користувачів:

- ціна;
- постачальник;
- взаємодія з MS SQL Server.

В таблиці 4.2 наведено аналіз альтернатив щодо відповідності заявленим критеріям.

Таблиця 4.2 – Аналіз альтернатив сховищ даних

Критерій	Файлова система на хостингу веб-рішення	Azure Blob	Amazon S3
Ціна	Включена в плату за хостинг	0.195 \$/Гб/міс + операції доступу	0.09 \$/Гб/міс + операції доступу
Постачальник	Хостинг-провайдер	Microsoft	Amazon
Взаємодія з MS SQL Server	Так	Так	Так

Зважаючи на ціни та вимоги до системи доцільно використовувати доступне файлове сховище на серверах хостинг-провайдера. Проте враховуючи популярність хмарних технологій окремі клієнти можуть отримати доступ до Azure Blob за умови оплати цих послуг.

4.3 Вибір технологій для серверного застосування

Вибраний перелік продуктів дозволяє розпочати аналіз технологій доступних для проектування серверного застосування. Критеріями вибору мають стати:

- кросплатформенність;

- доступними бібліотеками для роботи з Excel файлами;
- можливістю побудови API;
- висока швидкодія;
- робота з MS SQL Server.

Серед доступних технологій для вибору запропоновано наступні відомі та популярні рішення:

- Python Django;
- Node.js;
- ASP.NET Core.

Аналіз відповідності вибраних технологій критеріям наведено в таблиці 4.3.

Таблиця 4.3 – Аналіз відповідності серверних технологій критеріям

Критерій	Python Django	Node.js	ASP.NET Core
Кросплатформенність	Linux подібні OS	Так	Так
Excel файли	Доступні	Доступні	Доступні
API	Так	Так	Так
Робота з MS SQL	Так, через сторонні бібліотеки	Так, через сторонні бібліотеки	Так

Ключовим критерієм вибору виступає простота та зручність взаємодії з MS SQL Server. По цьому критерію значно краще виглядає ASP.NET Core через доступність цієї взаємодії по замовчуванню. Окрім прямої підтримки T-SQL запитів через ADO.NET також є доступними кілька ORM які будуть проаналізовані пізніше. Ще одним важливим кроком у виборі технологій реалізації є визначення необхідності використання ORM. Доступними для C# технологіями є:

- Entity Framework;
- NHibernate;
- SubSonic.

Окрім них варто виділити не ORM а технологію доступу до даних ADO.NET вбудовану в .NET Core.

ADO.NET дозволяє виконувати SQL запити, запускати процедури та повноцінно взаємодіяти з базою даною через SQL.

Ключовим недоліком NHibernate є недостатня підтримка LINQ, що, незважаючи на всю гнучкість, робить використання технології менш зручним у порівнянні з Entity Framework. А недоліком SubSonic є його невідомість та поширеність у вузьких кругах, що не може гарантувати коректної роботи та швидкої реакції на запити підтримки.

Перевагою ORM над використанням ADO.NET є пришвидшення розробки та інкапсуляція великого обсягу логіки. Крім того розробники технології вже виправили велику кількість помилок з якими стикаються під час використання прямих запитів до БД з коду. Ще одним важливим пунктом варто відзначити, що Entity Framework дозволяє виконувати (за необхідності) SQL запити.

У результаті аналізу та порівняння існуючих альтернатив на всіх рівнях вибраний стек технологій для розробки серверної частини системи виглядає так:

- MS SQL Server;
- Entity Framework;
- C#, ASP.NET Core;
- Web API;
- файлове сховище на серверах хостинг-провайдера (за вимогою Azure Blob).

4.4 Вибір технологій для реалізації веб-застосування

Набір технологій для розробки клієнтського застосування залишено стандартним для веб-застосувань:

- HTML 5;
- CSS 3;
- JS фреймворк.

5. РОЗРОБЛЕННЯ СТРУКТУРНОЇ СХЕМИ СИСТЕМИ

5.1 Загальна структурна схема

На основі вибраних технологій для розробки системи розроблено її структурну схему. На структурній схемі виділено такі рівні:

- сховища даних;
- бізнес логіка серверного рішення;
- веб-додаток серверного рішення;
- веб-клієнт та сторонні рішення.

На структурній схемі також виділено механізми взаємодії зі сторонніми рішеннями та основні структурні елементи кожного з перелічених рівнів.

Розробку системи було розпочато з формування структури бази даних. ER діаграму та інші схеми представлено та описано в окремому розділі. Як видно на нижченаведеній діаграмі на рівні зберігання даних окрім бази даних є ще два варіанти сховища файлів. Файлову систему сервера та Azure Blob було вибрано раніше. Обидва варіанти мають приблизно однакову структуру, елементи якої називаються по різному. Тут і далі «директорія» та «blob-об'єкт» з Azure Blob будуть називатися «текою» та «файлом», відповідно. Оскільки Azure Blob виступає лише додатковим сховищем даних у ньому буде реалізована ідентичну структуру розміщення файлів.

Відповідно до вимог документи та файли можуть бути завантажені в тему курсу. Їх розміщення в одній теці недоцільно в зв'язку з недопустимістю змішування файлів різних викладачів і з необхідністю подальшого розширення. Для цього розроблено наступну структуру файлового сховища (рис. 12).

Варто зазначити, що така структура сприяє подальшому розширенню системи, дозволить в майбутньому безперервно впровадити додаткові функції (пр. завантаження звітів по темі).

Аналогічну структуру використано для Azure Blob сховища, це дозволяє знатно зменшити повторюваність коду та складність логіки роботи з методами зчитування та запису файлів.

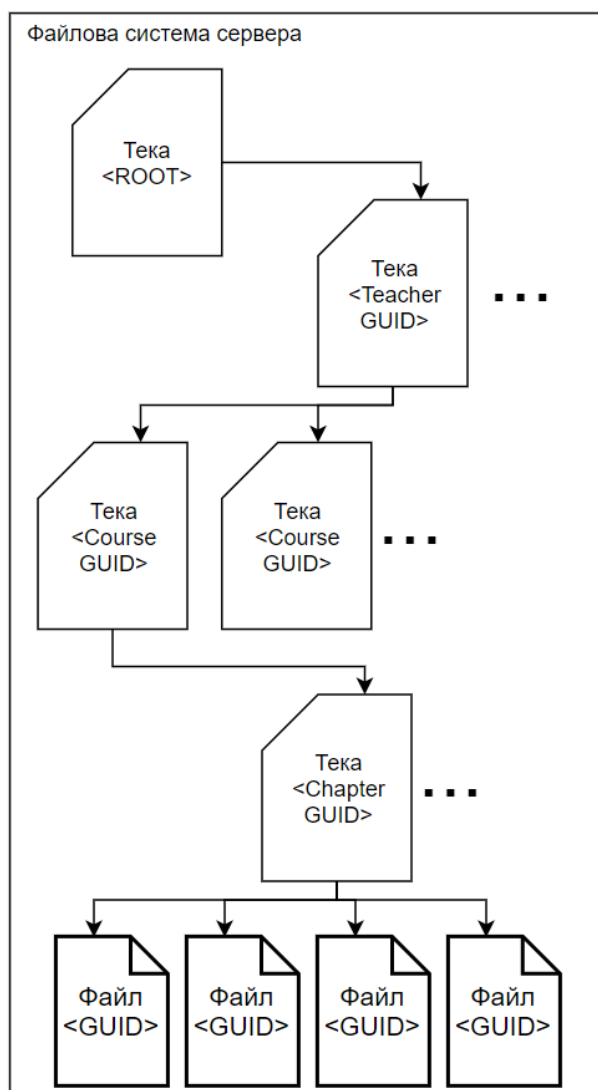


Рисунок 12 – Структурна схема файлової системи сервера

Окрім рівня зберігання даних на загальній структурній схемі (рис. 13) зображено структуру серверного рішення та веб-застосувань клієнтів. Нижче детально описані важливі елементи зображених модулів з розкриттям їх структури.

Рівень доступу до даних забезпечує систему методами, що дозволяють оперувати даними. Деталі реалізації буде розкрито в розділі «Реалізація бізнес-логіки системи».

Виділені структурні елементи рівня доступу до даних:

- ORM;
- модуль роботи з файлами.

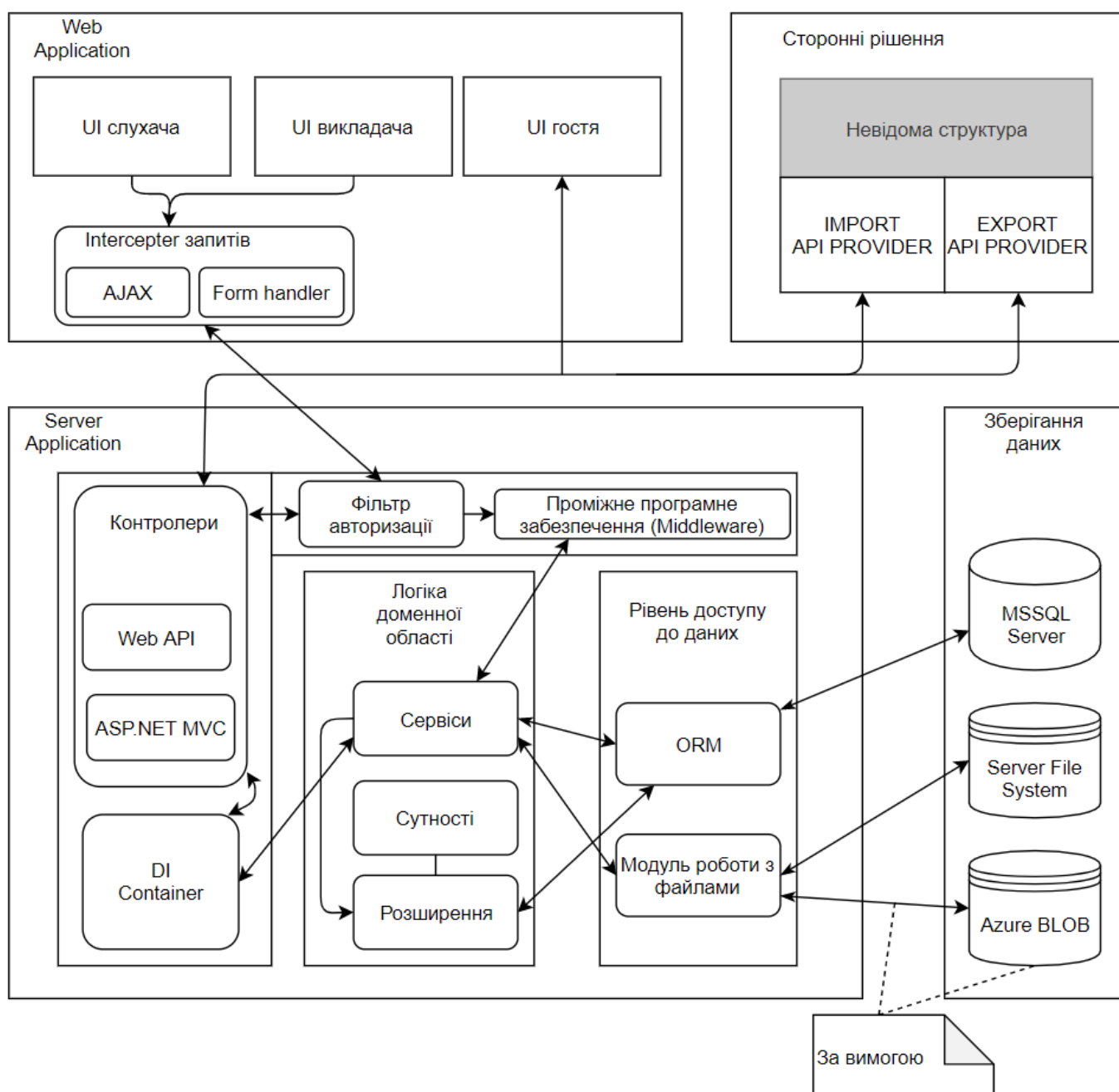


Рисунок 13 – Загальна структурна схема системи

Відповідно до раніше проведеного аналізу в якості ORM використано Entity Framework. В документації до фреймворку можна детально ознайомитися з його структурою.

5.2 Детальний огляд структурних елементів

Модуль роботи з файлами розроблено з дотриманням вимог розширюваності та зі забезпеченням мінімальної зв'язаності елементів. Детальну структуру модуля роботи з файлами зображено на рисунку 14.

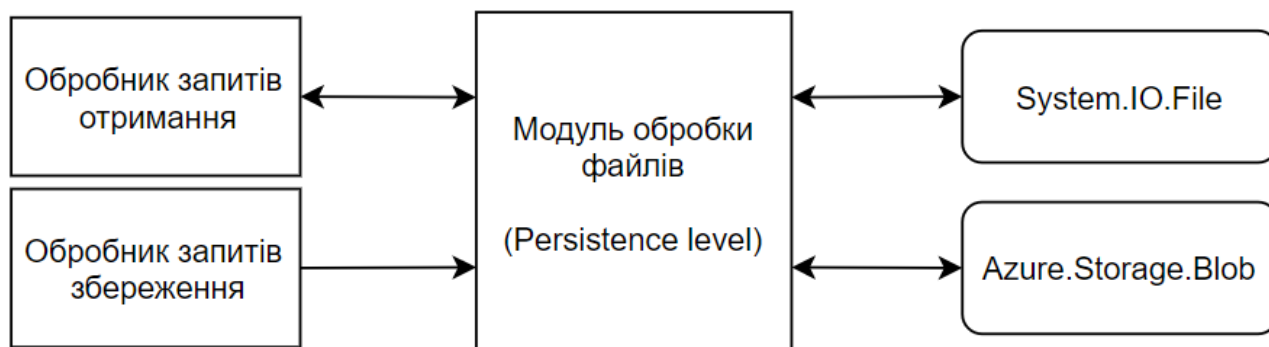


Рисунок 14 – Структурна схема модуля роботи з файлами

Рівень логіки доменної області цікавий своєю внутрішньою структурою. На цьому рівні система оперує бізнес-правилами, сутностями, зв'язками та саме тут виконується обробка запитів і формуються відповіді. Окрім наведених на загальній схемі структурних елементів виділено такі інкапсульовані члени:

- EPPlus;
- Platform.ImportExportSpreadsheet;
- Platform.Core;
- Platform.Domain.

Детальну структурну схему рівня логіки доменної області наведено на рисунку 15.

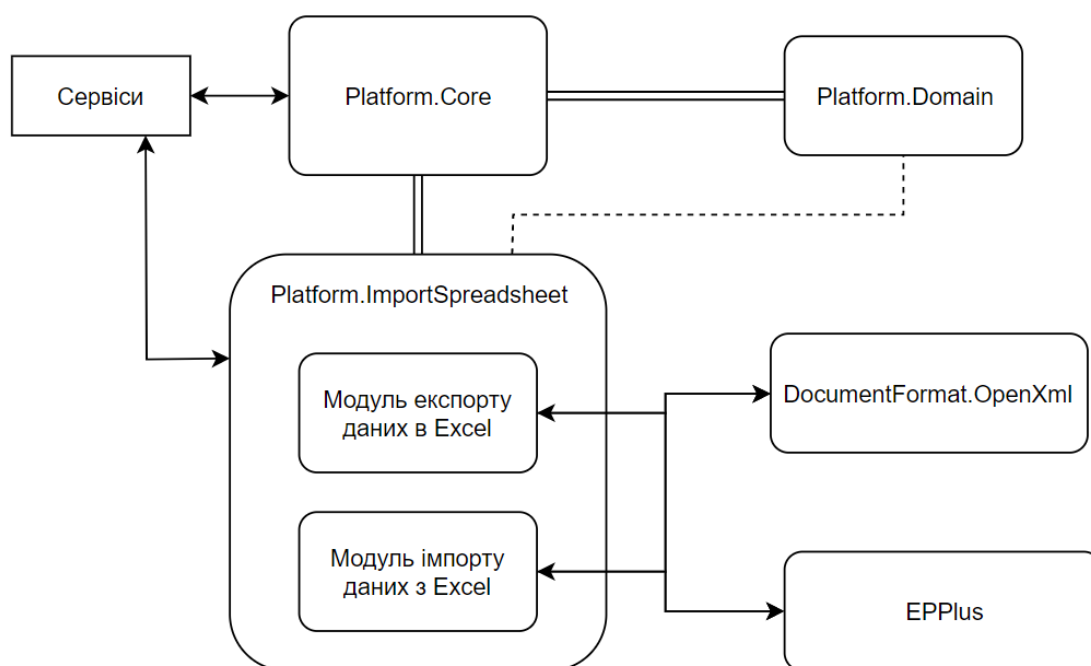


Рисунок 15 – Структурна схема рівня логіки доменної області

В бібліотеку окремо виділено модулі експорту даних в Excel файл та імпорту сутностей із Excel файлу. Крім того файли-шаблони зберігаються на рівні цієї бібліотеки та постачаються у формі вбудованого в збірку ресурсу.

Деталі реалізації всіх бібліотек та рівнів описано в наступних розділах.

На загальній структурній схемі присутні елементи які повинні забезпечуватися сторонніми розробниками для подальшого імпорту їх курсу в систему. На схемі це такі елементи:

- Import API Provider;
- Export API Provider.

Їх структура не може бути описана детально, оскільки розроблятиметься саме постачальниками стороннього курсу. Збоку системи видно, що взаємодія відбувається через API, проте детально не розкрито структуру модуля, який система використовує для здійснення імпорту та експорту. За виконання цих функцій відповідає модуль імпорту, який інкапсульовано в контролерах а бізнес-логіка роботи якого рознесена по сервісах.

Детальну структуру модуля імпорту зображено на рисунку 16.

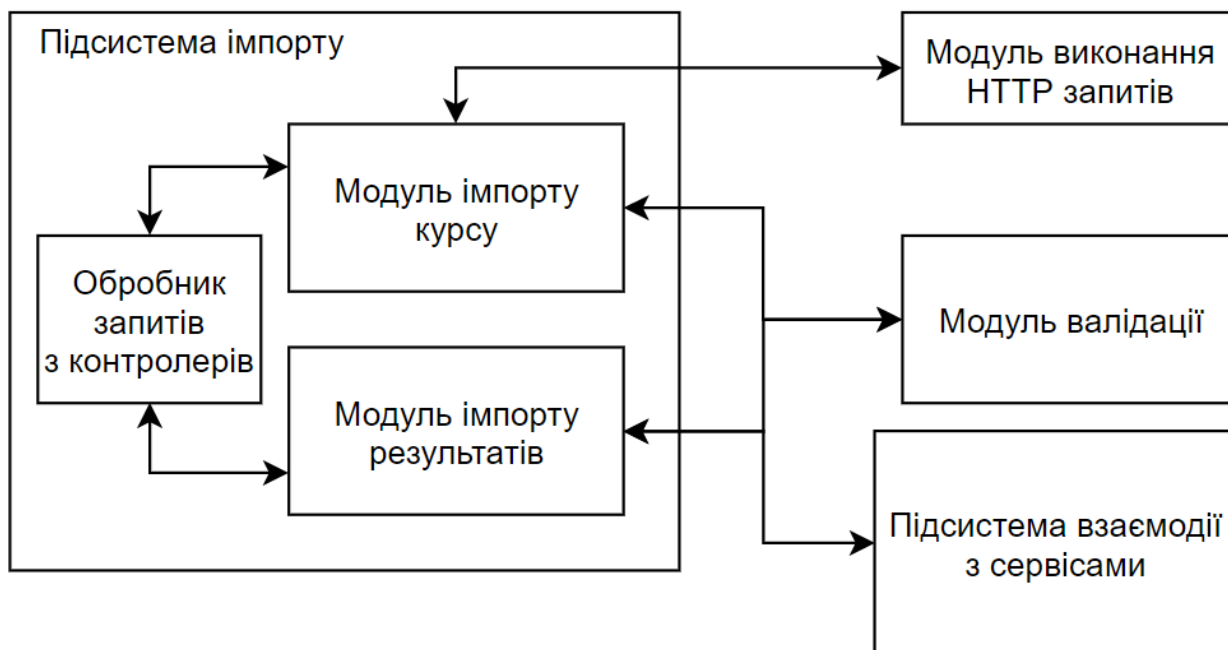


Рисунок 16 – Структурна схема підсистеми імпорту

Серед всіх елементів даної структурної схеми варто звернути увагу на модуль валідації. В контексті описаної підсистеми даний елемент відповідає за підтвердження можливості інтегрувати курс у систему, що означає гарантію безперебійної роботи функцій системи, що пов'язані з цим стороннім курсом. Деталі реалізації, шаблони та правила описано в наступних розділах.

Підсистему взаємодії з сервісами розроблено з метою інкапсуляції методів у сервісах, які не є важливими та не будуть використані під час імпорту курсу або результатів.

Таким чином у розділі повністю розкрито структуру системи в загальному та наведено структурні схеми елементів, ознайомлення з деталями яких необхідне та достатнє для розуміння роботи системи в цілому.

Повна, деталізована структурна схема системи доступна для ознайомлення в додатку Б.

6. РОЗРОБЛЕННЯ ER-ДІАГРАМИ

6.1 Інфологічна модель

Відповідно до вимог та опису прецедентів визначено необхідні для проектування сутності та зв'язки між ними. На ER-діаграмі зображено такі сутності:

- користувач;
- група;
- роль;
- курс;
- тема;
- наповнення «посилання»;
- наповнення «теорія»;
- наповнення «документ»;
- тест;
- завдання до тесту;
- варіант відповіді;
- відповідь користувача;
- дія користувача;
- результат користувача.

На рисунку 17 відображено інфологічну модель, яка включає список атрибутів сутності користувач.

Окрім сутності «Користувач» на діаграмі зображено також сутності «група» та «Роль». Варто зазначити, що з боку системи користувач – це слухач і викладач. Відповідно, відношення між деякими сутностями було розширено для відповідності вимогам. Відношення між наведеними на діаграмі сутностями наступні:

- користувач-група. Користувач може належати до однієї або не належати до жодної групи. Викладач завжди не належить до групи.

- користувач-роль. Користувач може бути в багатьох ролях (відповідно до вимог активна роль «Викладач», якщо вказана), при цьому, щонайменше одна роль обов'язкова.

На інших діаграмах атрибути раніше наведених сутностей не наводяться для уникнення неконтрольованого розростання діаграм.

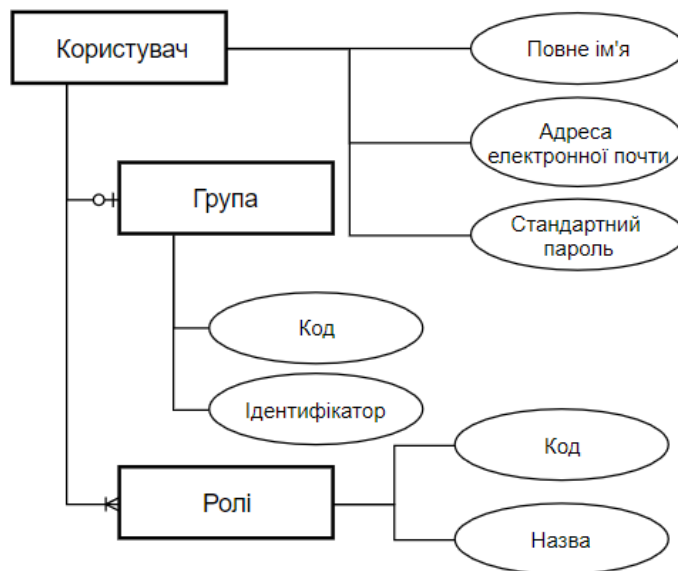


Рисунок 17 – Інфологічна модель сутності «Користувач»

Ключовим елементом системи після користувача, відповідно до вимог, виступає курс.

Атрибути та зв'язки сутності наведено на рисунку 18. На наведеній діаграмі сутність користувач розділено на викладача, який виступає власником курсу та слухача відмінності між зв'язками є та описуються так:

- викладач-курс. Викладач створює курс і володіє ним, курс не може існувати без викладача. Один і тільки один викладач є власником курсу. У викладача може бути багато курсів або жодного.
- Слухач-курс. Слухач має доступ до курсів. Щоб курс був доступним у нього має бути хоча б один слухач. Кожен слухач може мати доступ до багатьох курсів або до жодного. Кожен курс може бути доступним для багатьох користувачів.



Рисунок 18 – Інфологічна модель сутності «Курс»

Призначення атрибутів курсу відповідає опису полів з розділу «Формування вимог». Курс поділяється на теми, кожна з яких має свої атрибути. Зв'язок курс-тема один до багатьох з наступними обмеженнями:

- кожна тема може міститися в одному і тільки одному курсі;
- курс може складатися з багатьох тем, але на етапі створення може не містити жодної теми.

Окрім основних атрибутів сутність «тема» має зв'язки з сутностями наповнень, відповідно до вимог ці зв'язки такі:

- тема-посилання. В кожную тему може бути додано багато посилань, посилання не може існувати поза темою. Посилання є необов'язковим.

- тема-теорія. Тема може один і тільки один блок теорії, проте на етапі створення допускається відсутність цього блоку. Теорія не може існувати поза темою або дублюватися.
- тема-документ. До кожної теми може бути прикріплено кілька файлів, жоден файл не може існувати поза теорією, для кожної теми у випадку дублювання має створюватися власна копія файлу. В тему може бути не додано жодного файлу.

Окрім наведених зв'язків сутність «Тема» пов'язана з сутністю «Тест». Модель тесту та його зв'язки зображено на рисунку 19.

Відповідно до вимог система підтримує різновид питань у яких немає правильної відповіді, правильною в такому випадку вважається відсутність відповіді на питання.

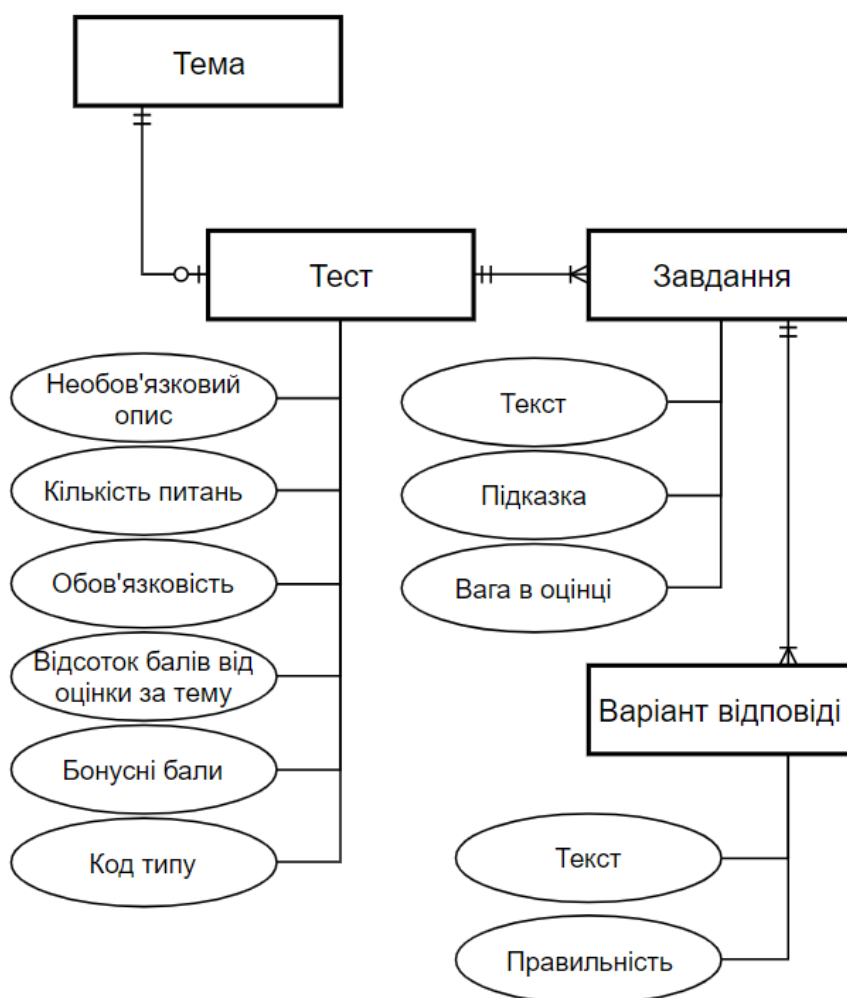


Рисунок 19 – Інфологічна модель сутності «Тест»

Дану модель побудовано відповідно до поставлених вимог, зв'язки між сутностями відповідають визначеним критеріям. На діаграмі описано такі зв'язки:

- тема-тест. Тема може містити тип наповнення «тест», кожен тест є унікальним у межах системи та не може існувати поза темою. Таким чином тест повинен бути присвоєний у одну і тільки одну тему. Проте не кожна тема може мати тест.
- Тест-завдання. Зв'язок описано як «складається з». Тест складається з набору завдань та не може існувати без хоча б одного завдання. Кожне завдання є унікальним у межах систему та не може бути використане в багатьох тестах. Завдання не може існувати поза тестом.
- Завдання-варіант відповіді. Формат тесту є загальновідомим і відрізняється від завдань із розгорнутою відповіддю наявністю варіантів відповідей. Кожне завдання може містити багато відповідей, та для валідності повинне містити хоча б один варіант відповіді.

6.2 Модель сутність-зв'язок

Описані сутності дозволяють навести модель інших сутностей, технічних, які забезпечують повноцінне функціонування системи відповідно до таких вимог:

- відстеження дій слухача;
- перездача тесту без обмеження кількості раз;
- збереження результатів оцінювання;
- оцінювання на основі активності а не тестування.

ER-діаграма зв'язків між сутностями та технічними сутностями зображено на рисунку 20.

Відповідно до вимог усі дії з наповненнями теми повинні запам'ятовуватися для проведення аналізу. Окрім того, що дії з наповненнями реєструються в системі вони напряду впливають на результати слухача. На діаграмі видно, що сутність

«Результат» є агрегуючою та виступає своєрідним висновком який робиться на основі активності слухача.

Окрім прямих дій, які запам'ятовує система результати агрегують також перевірку відповідей користувача на тест.

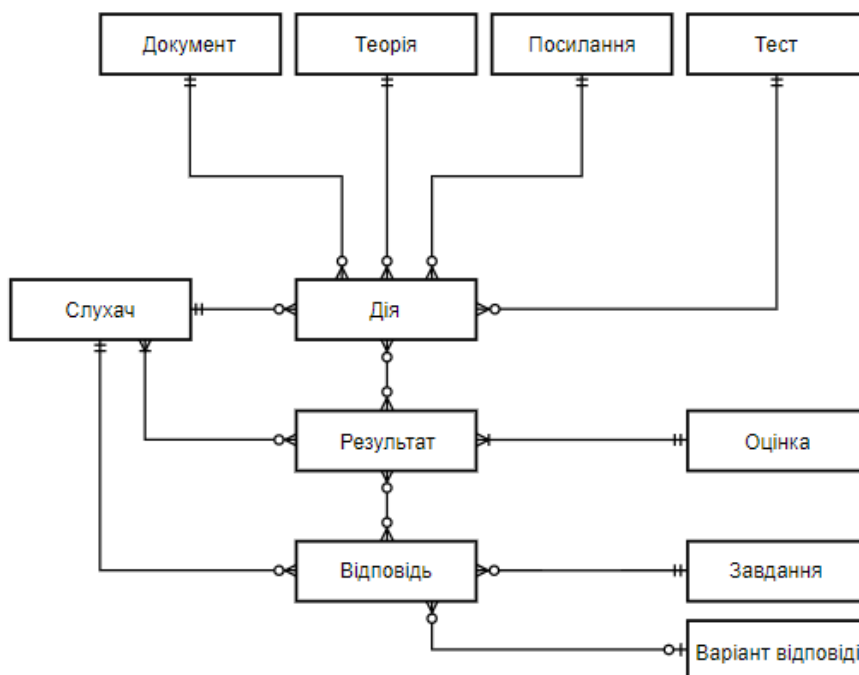


Рисунок 20 – ER-діаграма для технічних сутностей

Зв'язки між сутностями можна описати так:

- відповідь-завдання. На кожне завдання може бути дано багато або жодну відповідь.
- відповідь-варіант відповіді. Кожна відповідь супроводжується записом у базу даних варіанта відповіді, який дав користувач.
- відповідь-слухач. Кожна відповідь належить одному і тільки одному слухачу, слухач може не дати жодної відповіді.
- відповідь-результат. Відповіді формують багато результатів, якщо не дано жодної відповіді – не формується жодного результату.

Із загальною інфологічною моделлю можна ознайомитися в додатку В. На ній детально розкрито також атрибути сутностей, які не були наведені в даному розділі, та зображено усі зв'язки між усіма сутностями.

На основі розробленої ER діаграми вибрано первинні та зовнішні ключі для сутностей. Ці дані відображено в таблиці 6.1.

Таблиця 6.1 – Належність первинних та зовнішніх ключів до сутності

Сутність	Первинні ключі	Зовнішні ключі
Користувач	User Id	Group Id
Роль	Role Id	
КористувачРолі	User Id Role Id	User Id Role Id
Група	Group Id	
Курс	Course Id	User Id
Тема курсу	Chapter Id	Course Id Chapter Id Quiz Id
Тест	Quiz Id	Chapter Id
Завдання тесту	Question Id	Quiz Id
Варіант відповіді	Option Id	Question Id
Теорія	Theory Id	Chapter Id
Посилання	Link Id	Chapter Id
Документ	Document Id	Chapter Id

Визначивши первинні та зовнішні ключі таблиць розроблено таблиці БД для вибраних сутностей. В таблицях використовуються типи даних для стовпців, які є стандартними для СУБД MS SQL Server. Тип даних NVarchar – текст у форматі Unicode, використовується замість стандартного Varchar із метою повноцінної підтримки кирилиці.

6.3 Опис бази даних

Таблиця Users складається з таких стовпців (тут і далі формат у якому подано опис стовпців включає ім'я стовпця, тип даних(обмеження по довжині, якщо є), null сумісність):

- Id (PK), uniqueidentifier, не null;
- UserName, nvarchar(256), null;
- NormalizedUserName, nvarchar(256), null;
- Email, nvarchar(256), null;
- NormalizedEmail, nvarchar(256), null;
- EmailConfirmed, bit, не null;
- PasswordHash, nvarchar(max), null;
- FullName, nvarchar(max), null;
- DefaultPasswordHash, nvarchar(max), null;
- GroupCode, nvarchar(max), null;
- GroupId (FK), uniqueidentifier, null.

Ролі зберігаються в таблиці Roles, яка має наступну структуру:

- Id (PK), uniqueidentifier, не null;
- Name, nvarchar(256), null;
- NormalizedName, nvarchar(256), null.

Зв'язок між таблицями, відповідно до ER-діаграми багато-до-багатьох, реалізується проміжною таблицею UserRoles, що складається з таких стовпців:

- UserId (PK, FK), uniqueidentifier, не null;
- RoleId (PK, FK), uniqueidentifier, не null.

Окрім приналежності до ролей, користувачі так чи інакше належать до груп, які зберігаються в таблиці Groups. Таблиця групи проста та має лише такі стовпці:

- Id (PK), uniqueidentifier, не null;
- GroupCode, nvarchar(max), null;
- CreatedBy, nvarchar(max), null;

- DateCreated, datetimeoffset(7), не null.

Таблиця Actions описує дію користувача відповідно до вимог та ER-діаграми, структура таблиці така:

- Id (PK), uniqueidentifier, не null;
- ActionType, int, не null;
- DataId (FK), uniqueidentifier, не null;
- CreatedBy, nvarchar(max), null;
- DateCreated, datetimeoffset(7), не null.

Курси внесені в систему будь-яким з доступних способів зберігаються в таблиці Courses, яка складається з таких стовпців:

- Id (PK), uniqueidentifier, не null;
- CreatedBy, nvarchar(max), null;
- DateCreated, datetimeoffset(7), не null;
- Title, nvarchar(max), не null;
- Credits, int, не null;
- Hours, int, не null;
- Description, nvarchar(max), null;
- IsAvailable, bit, не null;
- IsExternal, bit, не null;
- ExternalUrl, nvarchar(max), null;
- OwnerId (FK), uniqueidentifier, не null.

Необхідно зазначити, що стовпець IsAvailable не означає відкритість для всіх слухачів, а відповідає за статус публікації курсу, якщо значення 0, курс не буде доступний жодному слухачу в системі, якщо 1 – курс відкритий для користувачів які додані в якості слухачів. За цей зв'язок відповідає таблиця UserCourseAccesses. Ця таблиця складається з таких стовпців:

- Id (PK), uniqueidentifier, не null;
- UserId (FK), uniqueidentifier, не null;
- CourseId (FK), uniqueidentifier, не null;
- CreatedBy, nvarchar(max), null;

- DateCreated, datetimeoffset(7), не null.

Як було відзначено раніше курс складається з тем, кожна тема є унікальна а тому описується окремим рядком в таблиці CourseChapters. Таблиця містить такі стовпці:

- Id (PK), uniqueidentifier, не null;
- CourseId (FK), uniqueidentifier, не null;
- CreatedBy, nvarchar(max), null;
- DateCreated, datetimeoffset(7), не null;
- Title, nvarchar(max), не null;
- Description, nvarchar(max), null;
- StaticPoints, int, не null;
- Order, int, не null;
- PreviewChapter (FK), uniqueidentifier, null;
- NextChapter (FK), uniqueidentifier, null;
- QuizId (FK), uniqueidentifier, null.

В описі таблиці Actions було вказано стовпець DataId. Він виступає нерегламентованим зовнішнім ключем на типи наповнень і те, значення ідентифікатора якого з типів у ньому записано залежить від значення у стовпці ActionType. Після опису таблиць типів наповнень буде розкрито залежність цих значень і посилання на тип.

Таблиця типу наповнення теорія називається TheoryAssignments і складається з таких стовпців:

- Id (PK), uniqueidentifier, не null;
- ParentId (FK), uniqueidentifier, не null;
- Type, int не null;
- IsRequired, bit, не null;
- PointsPercents, int, не null;
- PlainText, nvarchar(max), null;
- Html, nvarchar(max), null;
- CreatedBy, nvarchar(max), null;

- DateCreated, datetimeoffset(7), не null.

У таблиці наведено стовпець ParentId, який для теорії та для інших типів наповнення, що описано далі виступає зовнішнім ключем на таблицю CourseChapters.

Стовпці таблиці LinkAssignments, що описують тип наповнення посилання:

- Id (PK), uniqueidentifier, не null;
- ParentId (FK), uniqueidentifier, не null;
- Type, int не null;
- IsRequired, bit, не null;
- PointsPercents, int, не null;
- IsOuther, bit, не null;
- Text, nvarchar(max), не null;
- Url, nvarchar(max), не null;
- CreatedBy, nvarchar(max), null;
- DateCreated, datetimeoffset(7), не null.

В таблиці DocumentAssignment представлено метадані файлів, що завантажені для ознайомлення в межах теми. Схема таблиці наступна:

- Id (PK), uniqueidentifier, не null;
- ParentId (FK), uniqueidentifier, не null;
- Type, int не null;
- IsRequired, bit, не null;
- PointsPercents, int, не null;
- FileName, nvarchar(max), null;
- FileSize, nvarchar(max), null;
- FileId, uniqueidentifier, не null;
- CreatedBy, nvarchar(max), null;
- DateCreated, datetimeoffset(7), не null.

В цій таблиці видно кілька ідентифікаторів Id (PK) та FileId. Останній є неформальним ключем і саме по ньому надається доступ до файлу слухачу. Id (PK) тим часом, відповідає за запис дій користувача та пов'язані з цим події:

- нарахування балів за ознайомлення з файлами;

- видалення файлу з теми.

Як було сказано раніше в таблиці 6.2 наведено відповідність значень у стовпці ActionType посиланню на тип наповнення.

Таблиця 6.2 – Залежність зовнішнього ключа від типу дії

Значення у стовпці ActionType	Посилання на таблицю	Посилання на стовпець
1, 7	CourseChapter	Id
2	TheoryAssignment	Id
3	LinkAssignment	Id
4	DocumentAssginment	Id
5, 6	Quiz	Id
8	QuizTask	Id

Як видно з таблиці крім описаних раніше таблиць сутність, а отже й таблиця Actions має зв'язки з курсом та завданням курсу. Нижче наведено стовпці таблиці Quiz, яка відповідає за збереження даних тесту:

- Id (PK), uniqueidentifier, не null;
- ParentId (FK), uniqueidentifier, не null;
- Type, int не null;
- IsRequired, bit, не null;
- PointsPercents, int, не null;
- BonusPoints, int, не null;
- OptionalText, nvarchar(max), null;
- TasksCount, int, не null;
- CreatedBy, nvarchar(max), null;
- DateCreated, datetimeoffset(7), не null.

По аналогії з наповненнями ParentId – зовнішній ключ на CourseChapters. А первинний ключ таблиці Quiz є зовнішнім серед стовпців таблиці QuizTask:

- Id (PK), uniqueidentifier, не null;
- QuizId (FK), uniqueidentifier, не null;
- Text, nvarchar(max), не null;

- Tip, nvarchar(max), null;
- PointsWeight, int, не null;
- CreatedBy, nvarchar(max), null;
- DateCreated, datetimeoffset(7), не null.

Кожне завдання тесту має варіанти відповідей які зберігаються в таблиці QuizTaskOptions, стовпці якої наступні:

- Id (PK), uniqueidentifier, не null;
- QuizTaskId (FK), uniqueidentifier, не null;
- Text, nvarchar(max), не null;
- IsCorrect, bit, не null;
- CreatedBy, nvarchar(max), null;
- DateCreated, datetimeoffset(7), не null.

Таку структуру таблиць пов'язаних з курсом розроблено з метою забезпечення слухача можливістю проходити тест необмежену кількість раз, всі відповіді слухача зберігаються в таблиці Answers, яка має такі стовпці:

- Id (PK), uniqueidentifier, не null;
- QuizId (FK), uniqueidentifier, не null;
- QuestionId (FK), uniqueidentifier, не null;
- OptionId (FK), uniqueidentifier, null
- Text, nvarchar(max), не null;
- IsCorrect, bit, не null;
- CorrectOptionId (FK), uniqueidentifier, null;
- CreatedBy, nvarchar(max), null;
- DateCreated, datetimeoffset(7), не null.

На основі опису розроблено схему бази даних, яка доступна в додатку Г.

Варто також відзначити, що завдяки використанню ORM і СУБД MS SQL Server не було витрачено часу на налаштування предикатів цілісності, оскільки СУБД забезпечує цілісність на основі первинних та зовнішніх ключів, в результаті чого ситуація, коли в таблиці А є посилання на неіснуючий рядок в таблиці Б неможливо,

у випадку настання подібної ситуації – дія, що викликає її буде недопущено, а код, який викликав цю дію отримає повідомлення шляхом генерації виключення.

Окрім цього, засобами Entity Framework налаштовуються такі індекси:

- кластеризований, для первинних ключів типу `uniqueidentifier`;
- не унікальний, некластеризований для зовнішніх ключів зв'язку один-до-багатьох;
- унікальний, некластеризований для зовнішніх ключів зв'язку один-до-одного.

7. РЕАЛІЗАЦІЯ БІЗНЕС-ЛОГІКИ СИСТЕМИ

7.1 Вибір підходу та загальна архітектура системи

Після детального аналізу вимог, ознайомлення зі запропонованою структурою системи, технологіями, що були обрані для реалізації та на основі спроектованої бази даних лишилося відкритим питання проектування архітектури.

При виборі підходів до реалізації враховано такі критерії:

- забезпечення мінімальної зв'язаності [13];
- надання переваги композиції перед наслідуванням [14];
- зменшення повторюваності коду;
- перевикористання коду [15];
- зменшення витрат на хостинг за рахунок архітектури [16].

Першим ключовим рішенням обрано підхід серед популярних альтернатив:

- мікросервісна архітектура;
- монолітне рішення.

Незважаючи на можливість контейнеризації мікросервісів вирішальним аргументом проти вибору цього підходу стало збільшення вартості розміщення мікросервісів на серверах. Крім цього, після детального розбору вимог до системи та аналізу різниці в швидкодії обрано до розробки монолітне рішення. Діаграму розгортання представлено в додатку Д.

Очевидно, що для формування архітектури системи в цілому, розроблено декілька UML-діаграм, що розкривають деталі реалізації окремих компонентів.

Перед описом деталей реалізації виділено наступне:

- за основу взято поширену та популярну три-рівневу архітектуру з невеликими змінами, які впливають з вимог;
- використано поширені та популярні підходи до проектування рівня доступу до даних;
- бізнес логіку розподілено між сервісами та методами розширення;
- спроектовано репозиторій над контекстом бази даних.

На рисунку 21 зображено загальну архітектуру серверного рішення.

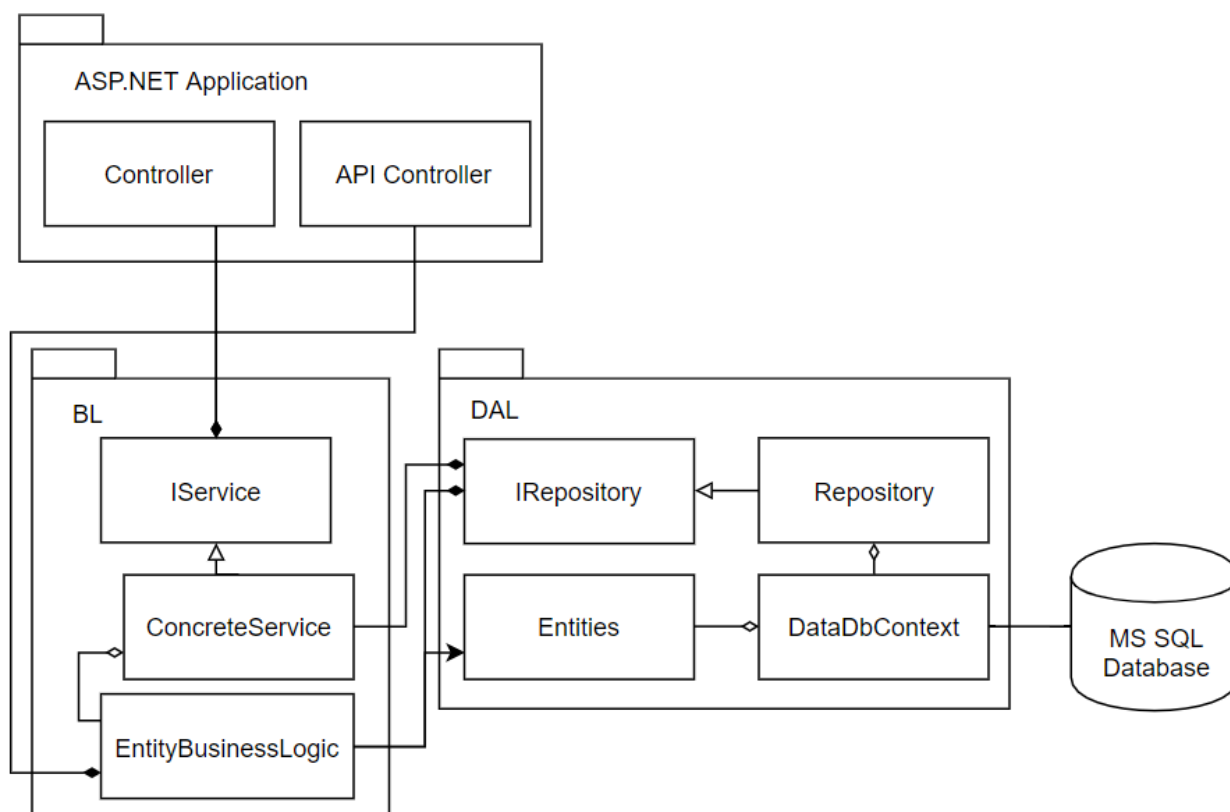


Рисунок 21 – Загальна архітектура серверного рішення

На діаграмі відображено взаємодію БД із рівнем доступу до даних, включення репозиторіїв до сервісів, та їх включення в рівень додатку.

На рисунку 22 зображено UML-діаграму поширеної реалізації шаблону «Репозиторій».

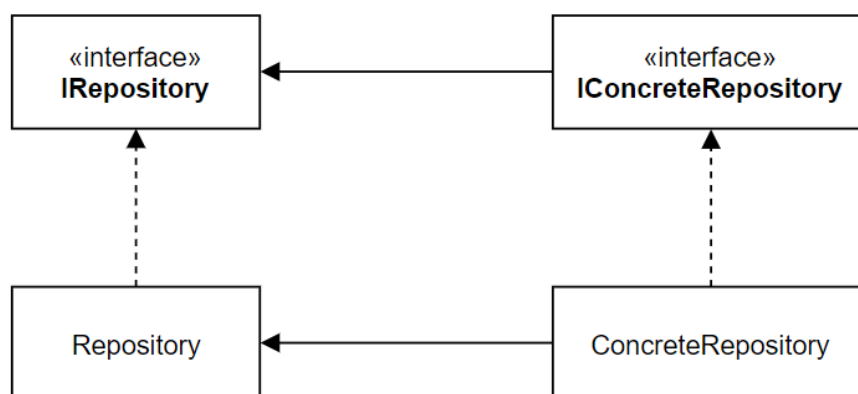


Рисунок 22 – Поширена реалізація шаблону «Репозиторій»

Ключовим недоліком цього підходу, не дивлячись на поширеність і наскрізні рекомендації в мережі Інтернет [17] у ході розробки архітектури системи, відповідно до поставлених критеріїв було розроблено наступну реалізацію шаблону «Репозиторій».

Окрім відмінностей з наведеним раніше шаблоном, реалізація запропонована на рисунку 23 дозволяє повною мірою використовувати можливості мови програмування C#.

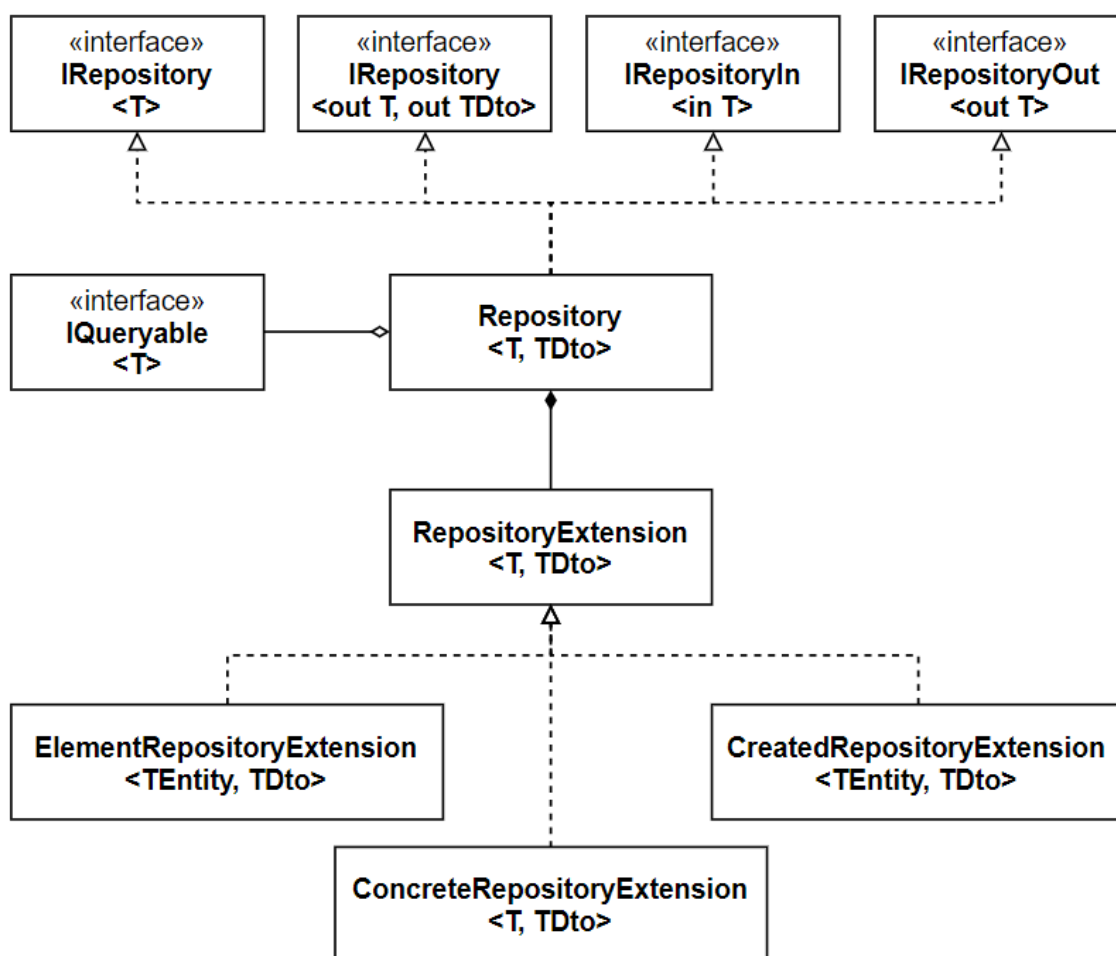


Рисунок 23 – Запропонована реалізація шаблону «Репозиторій»

Обґрунтування такого архітектурного рішення наступне:

- на основі аналізу розробленої структури бази даних визначено загальні для сутностей поля;

- уніфіковано тип первинного ключа для всіх сутностей та співставлено з представленим у C# типом даних;
- виділено поля, що відповідають за спільну функціональність та розроблено для них інтерфейсні типи;
- визначено типи запитів перед виконанням яких мають відбутися зміни у відповідних властивостях об'єктів, що реалізують інтерфейси.
- такий розподіл властивостей дозволив уникнути дублювання коду з логікою;
- такий підхід вирішує ключову проблему наведеного раніше варіанту реалізації шаблону – нагромодження та необхідності підтримки великої кількості конкретних репозиторіїв;
- розроблена реалізація дозволяє активно використовувати вбудований контейнер залежностей і такі засоби мови C# як методи розширення;
- не відбувається дублювання логіки при заповненні різних DTO із однієї доменної сутності.

Окрім наведених переваг у запропонованій реалізації шаблону «Репозиторій» активно використовується інтерфейс `IQueryable` та його узагальнена версія `IQueryable<T>`. За рахунок цього досягнуто таких переваг:

- методи LINQ, що викликаються на `IQueryable` транслуються в SQL засобами ORM EntityFramework [18];
- групування, фільтрація, з'єднання виконуються на рівні бази даних і без створення екземплярів об'єктів у пам'яті;
- залишається можливість матеріалізувати всю таблицю шляхом виклику метода `ToList()`.

Також, конкретно ця, вибрана реалізація дозволяє без істотних змін розширювати можливості системи за рахунок реалізації додаткових класів розширення репозиторію.

Як було показано на структурній схемі частину логіку роботи з Excel файлами (операції, які повинні підтримуватися системою описано у вимогах) винесено в окрему бібліотеку класів. Таке рішення прийнято з метою отримання можливості

розвивати та підтримувати цю бібліотеку без значного впливу на код всередині системи. Алгоритм імпорту даних з Excel таблиці вибрано наступний:

- завантаження файлу, збереження в теку тимчасових файлів;
- відкриття файлу для читання;
- шляхом виконання транзакції запис із рядків файлу в таблицю бази даних;
- запит на витягування всіх імпортованих рядків;
- створення облікових записів, розсилання повідомлень на емейл.

На рисунку 24 зображено UML-діаграму класів які використано у цій бібліотеці. На ній також розкрито ключові архітектурні рішення, що було прийнято в ході проектування цієї бібліотеки.

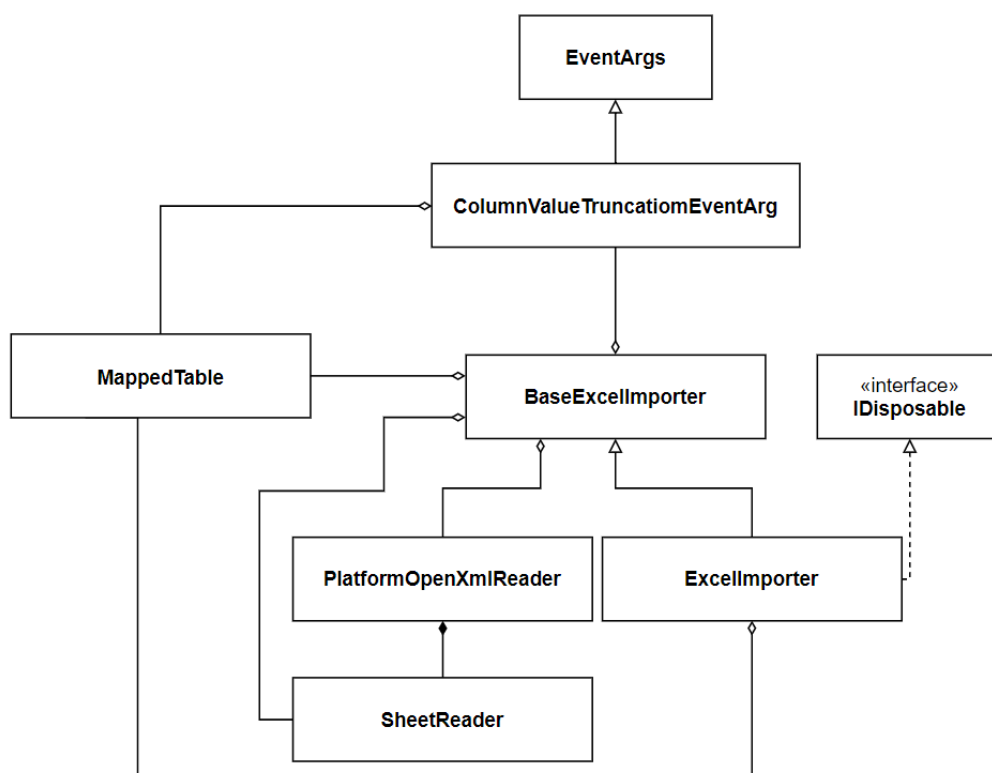


Рисунок 24 – UML-діаграма класів для імпорту з Excel файлу

Такий підхід дозволяє максимально розподілити логіку, відповідно до принципу одної відповідальності. За рахунок виділення окремих класів для запису рядків із файлу в базу даних, та виділення базового класу імпортера досягнуто значних успіхів у протидії дублюванню коду. Крім цього завдяки виділення класу

MappedTable досягнуто виділення однакового інтерфейсу для будь-якої кількості стовпців у ексель файлі та конфігурованості, що дозволило забезпечити простоту розширення та внесення змін у існуючі шаблони.

Крім імпорту з ексель файлу бібліотека також відповідає за експорт даних у Excel таблицю. На рисунку 25 представлено UML-діаграму класів, що відповідають за експорт даних у Excel файл.

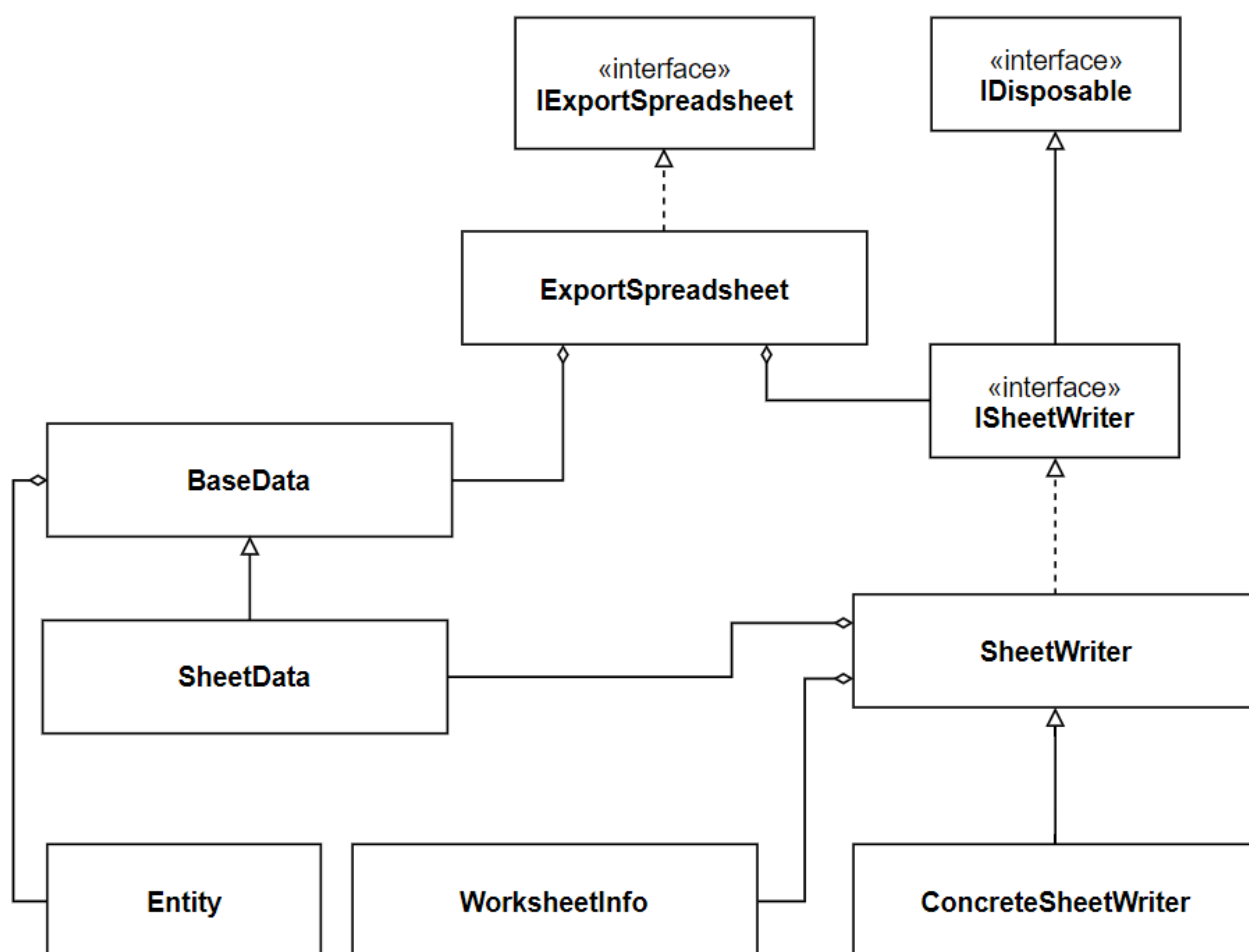


Рисунок 25 – UML-діаграма класів для експорту даних у Excel файл

Такі рішення прийнято для забезпечення простоти підтримки та розширення системи в подальшому. На діаграмі клас ConcreteSheetWriter відповідає за конкретну реалізацію логіки запису даних у таблицю. Для розширення бібліотеки іншими функціями експорту в ексель, достатньо створити новий клас Entity, який опише шаблон таблиці, створити нового наслідника SheetWriter і використовувати його

шляхом створення нового екземпляру узагальненого ExportSpreadsheet. Загальну архітектуру бібліотеки роботи з Excel наведено в додатку Е.

Відповідно до вимог система підтримує декілька типів наповнення теми курсу, після того як було досліджено схему бази даних, прийнято наступні архітектурні рішення:

- визначено спільні для всіх типів наповнення властивості;
- розроблено інтерфейс IAssignment, який забезпечить реалізацію властивостей;
- реалізацію додаткових членів покладено на класи, що реалізують інтерфейс;
- тема курсу містить не тільки загальний список IAssignment а ще й кожен тип наповнення окремо – рішення прийнято для забезпечення розробленої структури бази даних.

UML-діаграму класів описаних членів наведено на рисунку 26.

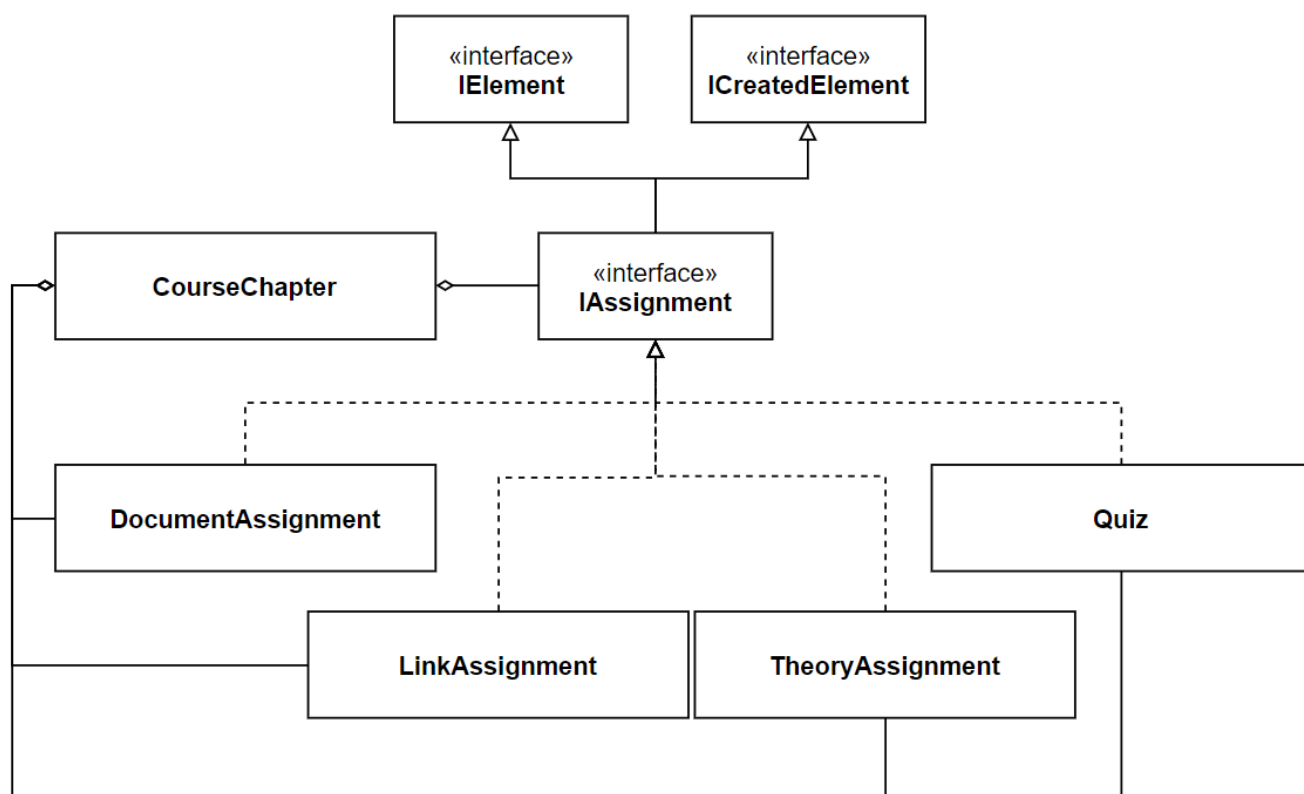


Рисунок 26 – UML-діаграма класів типів наповнення

Завдання які було вирішено таким чином:

- збереження структури бази даних із використанням підходу «спочатку код»;
- забезпечено обробку всіх типів наповнення шляхом роботи з визначеним інтерфейсом;
- дотримано принципу сегрегації інтерфейсів;
- всі типи наповнення гарантовано сумісні з реалізацією репозиторія;
- спрощено додавання нових типів наповнення;
- уникнено ускладнення логіки, що спростило підтримку.

Крім того у випадку впровадження додаткових балів за ті чи інші дії над окремим типом наповнення визначити максимальну кількість цих балів буде просто, оскільки з теми курсу є можливість отримати визначений тип наповнення.

Повну загальну архітектуру системи наведено в додатку Ж.

7.2 Розробка правил валідації

В зв'язку з тим, що для системи передбачено не тільки створення курсів із панелі керування курсами (вбудований механізм) а й імпорт сторонніх курсів, що зводиться до запиту на витягнення сторонніх метаданих (дані про курс сторонньої платформи, результати проходження курсу на сторонньому сервісі) – передбачено механізми перевірки коректності. Окрім цього, валідацію передбачено також для перевірки даних, які вносяться в систему шляхом імпорту файлів. Перед запуском курсу для викладача, доступна опція перевірки на коректність внесених даних.

Сутності, що представляють собою результати валідації описано діаграмою, зображеною на рисунку 27.

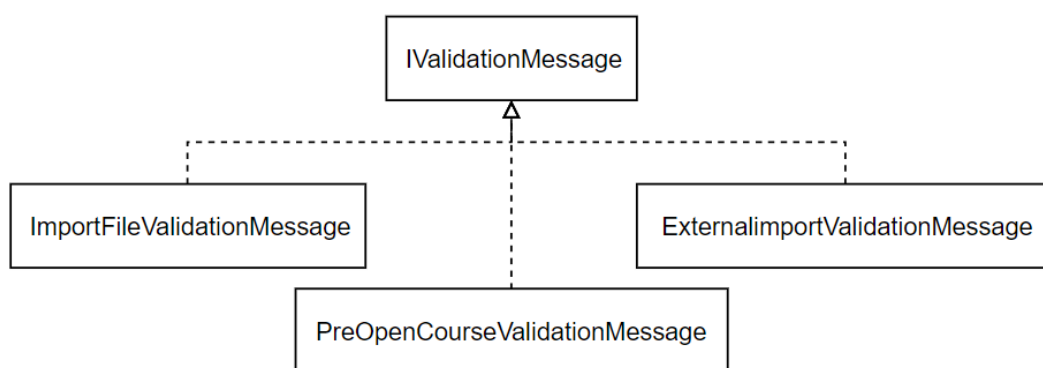


Рисунок 27 – Діаграма результатів валідації

Таку реалізацію розроблено в зв'язку з:

- в результаті валідації даних у файлі, що імпортується, має бути вказаний рядок на якому знайдено помилку;
- при виявленні помилки під час імпорту стороннього курсу, потрібно залогувати в системі в зв'язку з чим виникла ця помилка, для цього необхідно вказати властивості та значення які отримано від стороннього курсу;
- при валідації курсу перед відкриттям необхідно також представити додаткові дані про тему, наповнення, тест які є некоректними.

Правила валідації для імпорту даних із файлу процесу масової реєстрації представлено в таблиці 7.1.

Таблиця 7.1 – Правила валідації імпорту облікових записів

Назва стовпця	Ключ до ресурсів	Очікуване повідомлення	Опис
FullName	IsRequired	Значення в колонці FullName обов'язкове до заповнення	Обліковий запис без вказаного ПІБ не може бути створений у системі
FullName	SymbolsOnly	Недопустимі символи знайдено в колонці FullName	ПІБ може містити тільки буквенні знаки та пробіли
Group	IsRequired	Значення в колонці Group обов'язкове до заповнення	Група має бути заповнена. Помилка не критична.

Назва стовпця	Ключ до ресурсів	Очікуване повідомлення	Опис
Email	IsRequired	Значення в колонці Email обов'язкове до заповнення	В системі не може бути створено обліковий запис без адреси електронної пошти
Email	UniqueOnly	Обліковий запис з вказаним Email уже існує	В системі не може бути створено кілька облікових записів із однаковою адресою електронної пошти

У таблиці 7.2 наведено правила валідації для курсу створеного в межах системи, перед його запуском викладачем. У цьому випадку перевіряються на коректність не тільки метадані курсу, а враховуються також теми, наповнення та тест. Повідомлення показуються викладачу при явному виклику валідації.

Таблиця 7.2 – Правила валідації створеного в системі курсу

Сутність	Властивість	Очікуване повідомлення	Опис
Course	Title	Назва курсу є обов'язковою	Курс без назви не може бути запущений
Course	ChaptersCount	Жодної теми не додано	Пустий курс не може бути запущений
Course	ChapterPointsSum	Сума балів за всі теми перевищує максимальну N	Курс не може бути оцінений в більше балів ніж максимальне значення з конфігурації (крім бонусних)
CourseChapter	Title	Назва теми є обов'язковою	Тема без назви може некоректно відображатися слухачу
CourseChapter	Assignments	Тему не наповнено	Пуста тема недопускається

Сутність	Властивість	Очікуване повідомлення	Опис
Quiz	Questions	База запитань тесту менша ніж сконфігурована кількість запитань для слухача	Повторення запитань недопускається в тесті
TheoryAssignment	Html	Теорія обов'язкова	Пуста теорія недопускається

Імпорт даних із сторонньої системи перевіряється на відсутність помилок із максимальним рівнем критичності кожної неточності. Відповідно до вимог, структура відповіді для імпорту курсу до системи містить метадані, однією із полів якої є колекція метаданих усіх тем, які входять в курс.

Діаграму послідовності імпорту стороннього курсу наведено в додатку И. На ній можна прослідкувати коли відбувається валідація.

Правила валідації для імпорту стороннього курсу наведено в таблиці 7.3.

Таблиця 7.3 – Правила валідації для метаданих стороннього курсу

Сутність	Властивість	Повідомлення	Опис
StgCourse	Title	Поле Title не було заповнене постачальником курсу	Курс без назви може відобразитися некоректно.
StgCourse	ExternalId	Поле ExternalId не було заповнене постачальником курсу	За даним ідентифікатором відбувається імпорт результатів. Обов'язкове поле.
StgCourse	OpenUrl	Поле OpenUrl не заповнено постачальником курсу	Посилання є ключовим елементом для інтеграції курсу. Обов'язкове поле.
StgCourse	Credits	Формат даних невірний	Коректним значенням є ціле число в межах від 1 до 1000

Сутність	Властивість	Повідомлення	Опис
StgCourse	Hrs	Формат даних невірний	Коректним значенням є ціле число в межах від 1 до 1000
StgCourse	ChaptersMetadata	Жодної теми не надано постачальником курсу	Має бути надана хоча б одна тема
StgCourseChapter	Title	Поле Title не було заповнене постачальником курсу	Тема без назви може відображатися некоректно.
StgCourseChapter	ExternalId	Поле ExternalId не заповнено постачальником теми	За даним ключем відбувається прив'язка імпортованих оцінок до теми.
StgCourseChapter	PossiblePoints	Сума можливих балів за всі теми перевищує максимально можливу	Сума максимальної кількості балів за всі теми не має бути більшою за максимальну
StgCourseChapter	Description	Опис не заповнено	Опис обов'язковий

На основі розробленої архітектури, побудовано API доступу до методів серверного застосування та базовий інтерфейс користувача для демонстрації роботи системи.

8. РОЗРОБЛЕННЯ ІНТЕРФЕЙСІВ ВЗАЄМОДІЇ

8.1 API та документація

Для доступу до ресурсів, розроблено API. Початок роботи відбувається через виклик методу «domain/api/token» - цей метод дозволяє передати шляхом виконання запиту типу POST JSON об'єкт з полями:

- username, string, логін користувача зареєстрованого в системі;
- password, string, пароль користувача зареєстрованого в системі.

У відповідь, сервер поверне статус код OK або BadRequest, якщо дані вірні та користувач існує або якщо дані некоректні. Корисним навантаженням відповіді буде об'єкт JSON з полями access_token і username або message, відповідно.

Це єдиний метод API до якого мають доступ не авторизовані користувачі. Крім того ним можуть користуватися сторонні сервіси, які бажають, щоб слухачі їх курсів отримували доступ до системи шляхом використання аккаунтів із розробленої системи. Цей метод відповідає за аутентифікацію. Всі майбутні запити повинні супроводжуватися заголовком «Authorization» із значення «bearer {{access_token}}», де параметр access_token отриманий в результаті аутентифікації JWT токен.

В токен задовано роль користувача, в результаті чого відбувається розмежування доступу до ресурсів – студент не повинен і не може отримати доступ до дій викладача не знаючи його даних для входу.

Варто зазначити, що через токен відкривається повний доступ до системи і можливо отримати будь-які доступні дані або виконати дії (які будуть записані як активність). Крім того, спеціально для сторонніх систем, які будуть використовувати систему як платформу для авторизації розроблено метод «api/Account/UserInfo», GET запит на який без параметрів та тіла, зі заголовком авторизації дозволяє отримати дані облікового запису користувача. Рекомендується використовувати цей метод, для отримання ідентифікатора користувача в системі (у відповіді поле «id»). За рахунок цього імпорт результатів буде проведено за спрощеною процедурою (без співставлення електронних адрес та інших даних).

Всі методи API задокументовано шляхом генерації Swagger UI. Це забезпечує зручний інтерфейс взаємодії, та наочне представлення методів. Наприклад, на рисунку 28 наведено вигляд опису методів за шляхом «api/Account».

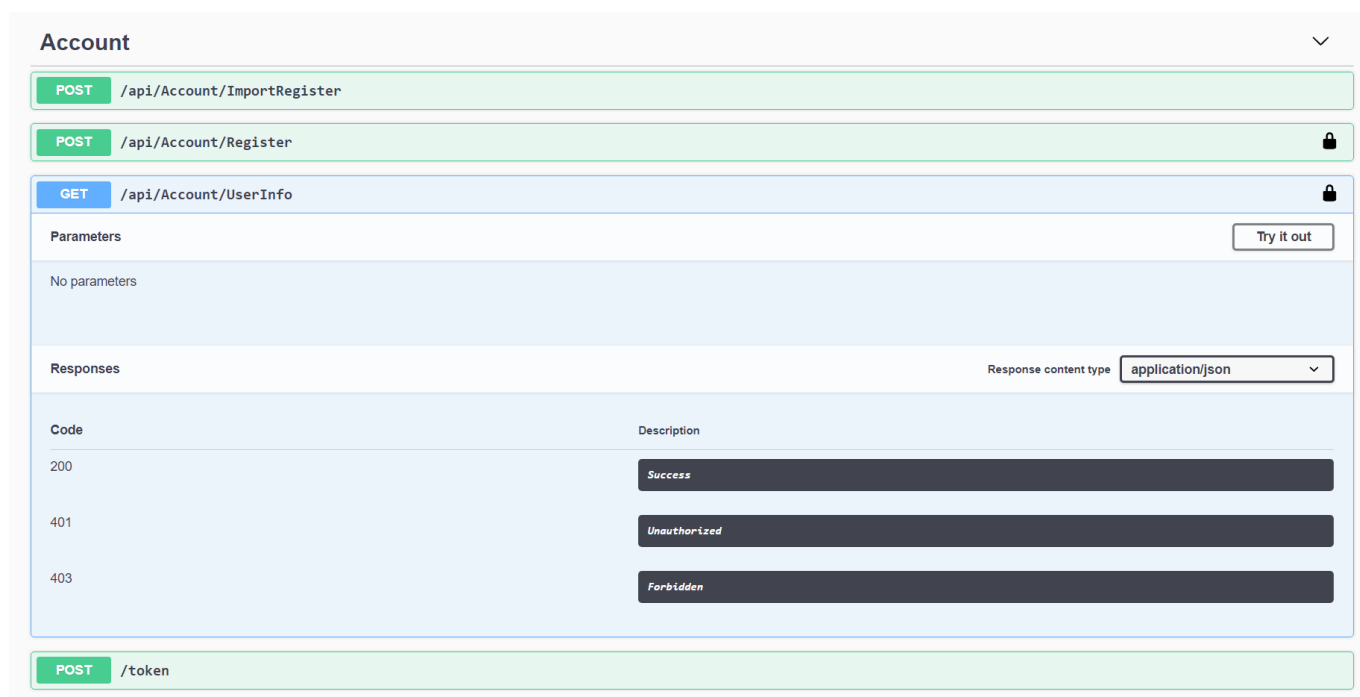


Рисунок 28 – опис методів за шляхом «api/Account»

Для запитів де необхідно передавати дані, або параметри URL рядка, UI представляє всі необхідні засоби. Поля додаються автоматично на основі моделей які отримують методи контролерів, які оброблюють запит.

На рисунку 29 зображено доступні методи для адміністрування курсу.

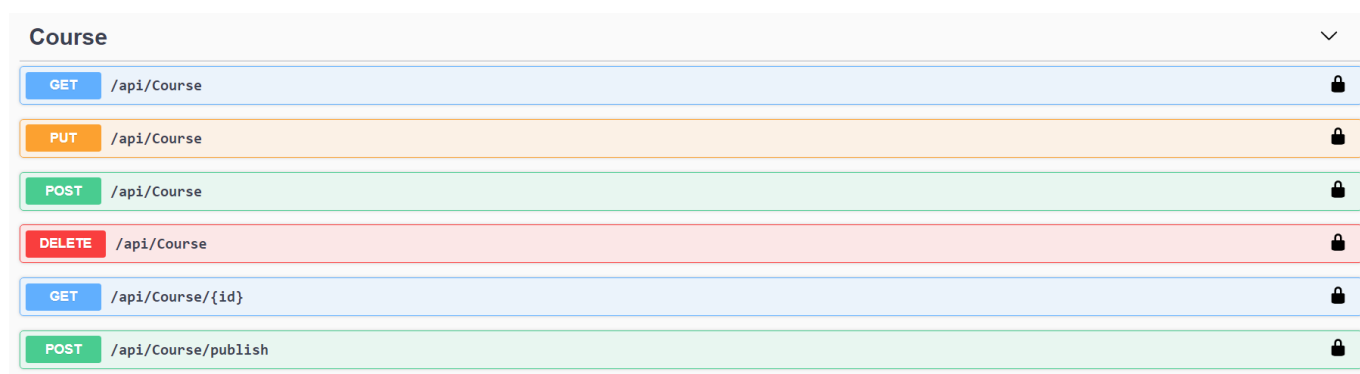


Рисунок 29 – Методи адміністрування курсів

Доступ до них мають тільки авторизовані викладачі а у випадку спроби доступу без коректного токена результатом запиту буде статус код 401 (Unauthorized). Крім того доступні методи адміністрування тем, які наведено на рисунку 30.

Chapters			▼
PUT	/api/course/{courseId}/Chapters		🔒
POST	/api/course/{courseId}/Chapters		🔒
GET	/api/course/{courseId}/Chapters/{id}		🔒

Рисунок 30 – Методи адміністрування тем

Для наповнення тем розроблено окремий блок методів, які об'єднані одним шляхом обов'язковим параметром якого є «ChapterId», оскільки таким чином відбувається прив'язка наповнення до теми. Ці методи зображено на рисунку 31.

Assignments			▼
PUT	/api/chapter/{chapterId}/Assignments/theory		🔒
POST	/api/chapter/{chapterId}/Assignments/theory		🔒
DELETE	/api/chapter/{chapterId}/Assignments/theory/{id}		🔒
PUT	/api/chapter/{chapterId}/Assignments/link		🔒
POST	/api/chapter/{chapterId}/Assignments/link		🔒
DELETE	/api/chapter/{chapterId}/Assignments/link/{id}		🔒
POST	/api/chapter/{chapterId}/Assignments/documents		🔒

Рисунок 31 – Методи керування наповненнями

Як було сказано у вимогах тест виділено окремим типом, для керування частинами тесту розроблено окремі блоки методів:

- для метаданих тесту;
- для запитань;
- для варіантів відповідей.

Ці блоки разом відповідають за контроль над тестом, всі їх перелічено на рисунку 32.

Quiz			▼
GET	/api/Chapter/{chapterId}/Quiz/{id}		🔒
PUT	/api/Chapter/{chapterId}/Quiz		🔒
POST	/api/Chapter/{chapterId}/Quiz		🔒
DELETE	/api/Chapter/{chapterId}/Quiz		🔒
QuizQuestions			▼
POST	/api/Quiz/{quizId}/QuizQuestions		🔒
DELETE	/api/Quiz/{quizId}/QuizQuestions		🔒
QuizTaskOptions			▼
POST	/api/QuizQuestions/{quizQuestionId}/QuizTaskOptions		🔒
DELETE	/api/QuizQuestions/{quizQuestionId}/QuizTaskOptions		🔒

Рисунок 32 – Методи керування тестом

Для того, щоб курс став доступний студенту необхідно вибрати цільову аудиторію, в якій буде доступ до конкретного курсу. За це відповідають методи наведені на рисунку 33.

CourseAccess			▼
POST	/api/CourseAccess/Group		🔒
DELETE	/api/CourseAccess/Group		🔒
POST	/api/CourseAccess/Personal		🔒
DELETE	/api/CourseAccess/Personal		🔒

Рисунок 33 – Методи керування доступом до курсу

Після цього всі необхідні для отримання інформації з системи методи, до яких мають доступ студент і викладач доступні через методи зображені на рисунку 34.

StudentCourses			▼
GET	/api/Student/Courses		🔒
GET	/api/Student/Courses/{id}		🔒
GET	/api/Student/Courses/{courseId}/Chapter/{id}		🔒

Рисунок 34 – Методи доступу до інформації для студента

Крім того розроблено також інші методи, які відповідають за збереження відповідей, трекінг дій тощо, проте вони не представляються у відкритій формі.

8.2 Інтерфейс користувача

Основними елементами з якими відбувається взаємодія виступають картки з метаданими про деяку сутність. Тут не наведено зображень загального вигляду інтерфейсу користувача а описано основні елементи з яких він складається.

Після авторизації в системі викладач бачить перед собою список карток із метаданими курсів які він створив або імпортував. Загальний вигляд карток наведено на рисунку 35.

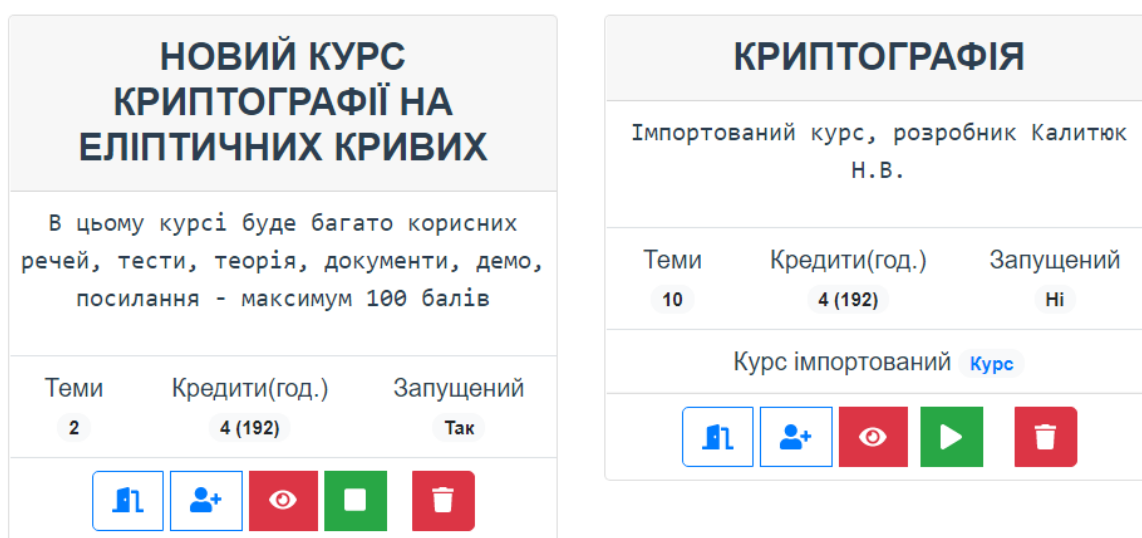


Рисунок 35 – Загальний вигляд курсів у списку доступних

Різниця між цими картками незначна, проте помітна. Блок із інформацією про те що курс імпортований та посилання на нього присутні тільки в сторонніх курсах. Незважаючи на те, що сторонній курс може бути активний викладачу необхідно надати до нього доступ (для доступу через систему) і запустити курс, щоб слухачі побачили його в списку доступних курсів. Кнопки запуску та зупинки курсу виглядають, як звичні елементи управління медіа (трикутник та квадрат).

Створення або імпорт курсу відбуваються на цій же сторінці шляхом використання форм зображених на рисунку 36.

Рисунок 36 – Форми для створення та імпорту курсу

Після створення курсу викладачу необхідно наповнити його темами, метадані тем також відображаються через візуальні та текстові блоки на картках. Загальний можливий вигляд огляду тем зображено на рисунку 37.

Рисунок 37 – Картки для огляду тем

Зображені іконки показують який тип наповнення вже додано в тему. Таким чином: книжка – означає, що теорію додано; посилання – тема містить посилання; скріпка – містить прикріплені документи; мозок – є тест.

Форма створення теми виглядає, як показано на рисунку 38.

Номер теми за порядком:

4

Назва:

Назва

Опис:

Опис курсу

Балів за тему:

15

Відмінити Створити тему

Рисунок 38 – Форма створення теми

Після створення теми доступне її відкриття для наповнення її інформацією. Відповідно до вимог повинні підтримуватися редактор теорії, створення та видалення посилань, завантаження файлів, створення або огляд тесту. Форми створення та огляду тесту наведено на рисунку 39.

Тест

Опис:

Бали (%):

Бонуси:

Завдань:

Тест

Оновлений тест для глави 4 (2)

Відсоток балів 45%

Бонусні бали 8

Завдань 10

База 4

Рисунок 39 – Форми створення та огляду тесту

Загальний вигляд форми для створення та редагування теорії зображено на рисунку 40.

Рисунок 40 – Загальний вигляд форми створення та редагування теорії

Форму керування посиланнями зображено на рисунку 41.

Рисунок 41 – Форма керування посиланнями

Як видно на зображенні всі кнопки мають спливаючі підказки, з описом дії, яку вони виконують. Ці підказки з'являються при наведенні на елемент управління. Крім того, в системі не передбачено підтвердження видалення, оскільки ця дія вимагає

подвійного натискання миші, що унеможлиблює імовірність випадкового видалення даних із системи.

На рисунку 42 зображено форму завантаження документів.

Документи

Назва файлу:

Відсоток балів:

Файл: No file chosen Browse

Завантажити

Вже завантажені

Дисертація.pdf (0,18 MB) 5%

Рисунок 42 – Форма завантаження документів

Незважаючи на те, що назву файлу можна взяти з даних про файл, часто є бажання, щоб студент бачив більш описову назву, задля цього додано поле «Назва файлу» яка буде відображатися в системі у слухача. Поле необов'язкове і якщо воно не заповнене, буде використано назву з даних про файл.

Інші елементи керування побудовано за ідентичним принципом, надлишково буде наводити їх усі.

Важливо зазначити, що інтерфейс дозволяє максимально просто та продуктивно адмініструвати електронні навчальні курси, як сторонні так і створені шляхом використання вбудованих можливостей системи. Можна зробити висновок, що розробку API та UI виконано успішно.

За рахунок використання компонентного підходу подальша розробка інтерфейсу користувача спрощена до рівня інтеграції та розміщення готових, розроблених компонентів. Крім того, ці ж елементи успішно перевикористовуються для відображення даних для слухача.

9. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

9.1 Опис ідеї проекту

Зміст ідеї: система адміністрування електронних навчальних курсів із розширеною функціональністю відстеження активності слухача для більш справедливого оцінювання, засобами розширення за рахунок імпорту електронних курсів створених сторонніми розробниками та наповнення через інтерфейс системи.

Напрями застосування: інтеграція в навчальний процес студентів денної форми навчання, організації навчальних курсів у слухачів, які навчаються віддалено (заочна форма, факультативні заняття). Збір інформації про зацікавленість слухачів у матеріалах курсу, реальний рівень мотивації оцінкою. Надання доступу окремим слухачам (групам, студентам) до матеріалів курсу (теоретичних відомостей, корисних посилань, сторонніх застосунків, PDF документів) та засобів перевірки знань за темами.

Вигоди для користувача: економія часу, доступ у будь-який зручний час, справедливе оцінювання, немає необхідності встановлювати програмне забезпечення.

В таблиці 9.1 наведено вигоди в залежності від типу користувача (викладач, слухач).

Таблиця 9.1 – Вигоди для користувача залежно від типу

Тип користувача	Вигода	Опис
Викладач	Економія часу при реєстрації	Імпорт облікових засобів шляхом завантаження файлу з визначеною структурою
Викладач	Розподілення курсів за інтересами	Готовий курс можна відкрити для доступу окремим слухачам або цілим групам
Викладач	Справедливе оцінювання	Завдяки розширеним можливостям відстеження активності слухачів поставити справедливу оцінку простіше

Викладач	Систематизація матеріалів по курсу	Матеріали курсу прив'язуються до конкретної теми, при посиланні на матеріалі можна одразу прикріпити джерело, це спростить пошук першоджерел
Викладач	Актуальність системи за рахунок розширюваності	Система розширюється за рахунок реалізації АПІ сторонніми розробниками, не потрібно змінювати платформу для включення додаткової функціональності (демо, парні завдання)
Викладач	Нецентралізоване впровадження	Для інтеграції системи не потрібно використовувати послуги посередників із налаштування та розгортання
Слухач	Доступ у будь-який час	Курс для проходження доступний цілодобово з моменту його відкриття до моменту поки слухача не буде видалено зі списку доступу
Слухач	Матеріали курсу систематизовані	Курс розділено на теми, матеріали курсу прив'язані до теми, це спрощує пошук інформації
Слухач	Справедливе оцінювання	За рахунок збору інформації про активність зменшується шанс отримати недостатньо високу оцінку для слухачів, які зацікавлені в курсі та докладають зусиль

Техніко-економічні характеристики ідеї: відстеження успішності проекту може відбуватися за різними критеріями залежно від обраної моделі розповсюдження. Для початкового аналізу обрано наступні характеристики:

- кількість впроваджень у викладачів;
- кількість курсів наповнених через систему;
- загальна кількість курсів;
- кількість слухачів;
- середня кількість звернень на 1 слухача;

- середня оцінка по групі;
- показник відношення кількості курсів до кількості викладачів.

Загальноекономічний аналіз: ринок дистанційного та змішаного навчання росте швидкими темпами, прослідковується тенденція зростання популярності електронних курсів та онлайн навчання. Крім того, очевидно, що вхід у сферу з орієнтованістю на слухача є недоцільним через такі фактори:

- існування аналогів, які вже мають тисячі курси від відомих викладачів;
- інертність слухачів, небажання зміни звичного інтерфейсу;
- витрати на розробку функціональності видачі сертифікатів, які неможливо підробити, можна інтегрувати до платформ пошуку роботи можуть зробити проект таким, що окуплятиметься довше;
- необхідно виділити значні кошти на рекламну кампанію для залучення викладачів і слухачів.

Переваги вибору орієнтації на викладача наступні:

- низький рівень конкуренції;
- авторські матеріали, що дозволить бути впевненим у якості курсу;
- впровадження аналізу активності слухача окрім справедливого оцінювання дозволить зібрати дані про «сезонну активність» слухачів і мотивуючі фактори;
- відкриває додаткові джерела монетизації.

9.2 Концепції конкурентів

Мій проект: система адміністрування електронних навчальних курсів з функціональністю орієнтованою на викладача.

Конкурент 1: платформи дистанційного навчання (Prometheus, Coursera, інші аналогічні) орієнтовані на слухача.

Конкурент 2: платформи розроблені для інтеграції в навчальний процес закладів освіти (Moodle, інші).

W – слабкі сторони:

- залежність від викладача;
- рівень технічної забезпеченості українських навчальних закладів;
- витрати на хостинг.

S – сильні сторони:

- відсутність аналогічної функціональності в конкурентів;
- можливість розробки додаткових функцій окремим закладам за вимогою;
- інтеграція в навчальний процес студентів денної форми навчання;
- простий, зрозумілий інтерфейс;
- відкритий API для розширення.

N – нейтральні сторони:

- слухачі, що використовують кілька платформ;
- веб-сервіси, що частково покривають функціональність системи.

Висновок: ідея проекту актуальна, чітко виокреслюються вигоди для викладачів і слухачів електронних навчальних курсів, крім того наведені сильні, слабкі та нейтральні сторони явно вказують на те, що проект може бути успішним і займе обрану нішу.

9.3 Технічний аудит ідеї проекту

Для технічного аудиту ідеї функціональність проекту, технології реалізації, наявність технологій та їх доступність представлено в таблиці 9.2.

Таблиця 9.2 – Технічний аудит ідеї проекту

Ідея проекту	Технологія реалізації	Наявність технологій	Доступність технологій
Зберігання матеріалів	СУБД MS SQL Server	Наявна	Доступна
Сховище файлів	Сервер системи, Azure Blob, Amazon S3	Наявна	Доступна

Реєстрація через імпорт	MS Excel	Наявна	Доступна
Відстеження активності слухача	Засоби ASP.NET	Наявна	Доступна
Програмування серверного рішення	Мова програмування C#, фреймворк ASP.NET Core	Наявна	Доступна
Відкрите API	ASP.NET Web API	Наявна	Доступна
Розширення сторонніми рішеннями	Засобами API	Наявна	Доступна
Клієнтське веб застосування	VueJS, Angular, HTML, CSS, JS	Наявна	Доступна
Обрані технології реалізації проекту: мова програмування C#, фреймворк ASP.NET Core (в т.ч. Web API), VueJS			

Висновки: необхідні технології вже існують, доступні для застосування безкоштовно або за несуттєву плату, враховуючи це проект може бути реалізований через використання обраних технологій.

9.4 Аналіз ринкових можливостей запуску проекту

9.4.1 Попередня характеристика потенційного ринку стартап проекту

Показники стану ринку та відповідні їм характеристики наведено в таблиці 9.3.

Таблиця 9.3 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку	Характеристика
1	Кількість головних гравців	4 (викладач, слухач, розробник, конкуренти)
2	Загальний обсяг продажу бізнес-продуктів	Понад 5000
3	Динаміка ринку	Зростає
4	Наявність обмежень для входу на ринок	Існують великі гравці, що частково покривають вимоги ринку, присутні дрібні гравці

5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності по галузі або по ринку у відсотках	90%

Висновки: На сьогодні банківський відсоток по вкладу в гривні складає в середньому 15%. Це приблизно в 6 разів менше за рентабельність даного проекту, очевидно, що є сенс вкладати гроші в проект. Крім того динаміка ринку показує, що він зростає – це робить його привабливим для входу. Незважаючи на великих гравців і велику кількість дрібних конкурентів проект займе вільну або низькоконкурентну нішу на ринку.

9.4.2 Характеристика потенційних клієнтів стартап проекту

Аналіз потреб що формує ринок, цільової аудиторії, поведінки аудиторії та вимог споживачів наведено в таблиці 9.4.

Таблиця 9.4 – Характеристика потенційних клієнтів

Потреби, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до бізнес продукту
Інтеграція онлайн-курсів у навчальний процес студентів денної форми, учнів, слухачів курсів змішаної форми навчання	1. Викладачі 2. Учні 3. Студенти 4. Інші зацікавлені у навчанні особи	Зручність інтерфейсу користувача. Вплив активності на кінцеву оцінку. Справедливе оцінювання. Ціна за одиницю бізнес-продукту. Підтримка та реакція на запити додаткової функціональності. Доступність в	До продукції: Обґрунтована ціна. Надійність. Якість. Зручність. Зворотній зв'язок. Простота інтеграції. До компанії - постачальника: Підтримка в робочий час. Швидке впровадження.

		цілодобовому режимі. Звичний доступ.	Відповідність заявленій функціональності. Реагування на запити змін або розширення.
Аналіз активності слухачів для справедливого оцінювання	1. Викладачі 2. Слухачі	Звітність про активність слухачів. Збір даних для забезпечення справедливого оцінювання.	До продукції: Відповідність заявленій функціональності. Частковий трекінг активності на сторонніх курсах. Простота в розширенні. До компанії-постачальника: Швидка реакція на запити підтримки. Можливість консультації, відповіді на запити документованих методів API.

Відповідно до вищенаведеного аналізу очевидно, що цільовими клієнтами є викладачі, слухачі електронних навчальних курсів, студенти денної та змішаної форми навчання, учні. Крім того клієнтами можуть бути власники стаціонарних курсів, репетитори.

9.5 Фактори загроз

В таблиці 9.5 наведено фактори, що перешкоджають ринковому впровадженню проекту.

Таблиця 9.5 – Фактори загрози

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Невідомість	Цільова аудиторія, зокрема викладачі не бажають витратити зусилля на наповнення платформи, про яку їм досі було невідомо контентом	Проведення презентації з демонстрацією можливостей системи на демо-курсі. Надати письмову гарантію.
2	Технічна орієнтованість	Неможливість наповнення курсів певного типу через засоби платформи	Надати знижку клієнту. Розробити сторонній курс, який буде імпортованим у загальну платформу.
3	Залежність від цін хостинг-провайдера.	Зміна хостинг-провайдера часто пов'язана з переносом існуючих даних у нову БД, вимкненням або некоректною роботою сервісів	Забезпечити часте резервне копіювання. Налагодити процеси доставки нових версій проекту засобами автоматизації. Повідомити про технічні роботи зарання. Проводити роботи в години меншої завантаженості.
4	Прив'язаність до вибраних технологій	Технології постійно розвиваються, при відмові в подальшій підтримці компанією-провайдером технологій клієнти будуть незадоволені відсутністю реакції	Обрати для побудови рішення технології які нещодавно вийшли на ринок, дослідити та обрати компанію-провайдера технологій, яка славиться довгим періодом підтримки своїх продуктів. Повідомити користувачам про

			використання технологій від компанії відомої у всьому світі.
5	Підтримка студентів	Прив'язаність наповнення матеріалами до викладача не забезпечує студента вільним вибором курсу для вивчення	Інтегрувати сторонні курси для загального доступу без прив'язки до викладача.
6	Сезонність активності	Онлайн курси мотивують слухачів сертифікатами та стиснутими термінами проведення курсів, під час інтеграції в навчальний процес закладів освіти електронні курси є допуском до заліку або проміжного оцінювання, критично великі навантаження будуть в період перед цим оцінюванням	Ввести систему додаткової мотивації для слухачів які вчасно та регулярно проходять курси. Впровадити функціональність нагадувань і повідомлень про терміни проходження курсів.
7	Недобросовісна конкуренція	Впровадження аналогічної функціональності більш відомим платформами-конкурентами	Постійна підтримка, розширення вже розроблених функцій. Інтеграція засобів машинного навчання.

Наведені фактори, очевидно, впливають на впровадження проекту а недобросовісна конкуренція може позбавити деякого відсотка клієнтів проте всі наведені загрози можуть бути зменшені або нівельовані за рахунок реакції компанії-постачальника, найбільша загроза – невідомість. Ефективним засобом боротьби з нею є проведення масштабної рекламної кампанії орієнтованої на викладачів та вчителів засобами персоналізації оголошень.

9.6 Фактори можливостей

В таблиці 9.6 наведено фактори, що сприяють ринковому впровадженню проекту.

Таблиця 9.6 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Популярність електронних навчальних курсів	Прослідковується тенденція зростання популярності онлайн-курсів та дистанційного навчання, що відкриває додаткові можливості розширення сфери застосування проекту	Окрім вибраного напрямку орієнтації на викладача додатково розвивати функціональність орієнтовану на слухачів і відокремленість курсу від закладу освіти в якому він проводиться
2	Охоплення аудиторії конкурентами	Слухачі електронних навчальних курсів використовують декілька платформ для онлайн-навчання, що дозволяє не переманювати слухачів у конкурентів, а бути ще одним проектом, яким користується клієнт	Проводити рекламну кампанію орієнтовану на розголос інформації про авторські навчальні від найкращих викладачів із провідних закладів освіти
3	Відсутність конкурентів орієнтованих на викладача	Платформи-конкуренти орієнтовані на слухачів і обрали модель монетизації за підпискою або плата за додаткову функцію	Якомога швидше зайняти нішу орієнтованих на викладача платформ, вчасно реагувати на пропозиції та відгуки тестової аудиторії
4	Інтеграція в навчальні заклади	Інтеграція рішення для одного викладача відкриває можливість впровадження для інших викладачів які вже будуть	Організувати демо версію, забезпечити безкоштовний доступ для обмеженої кількості викладачів і

		знайомі з системою та матимуть відгуки від колеги, яким можна довіряти	продавати додаткові облікові записи
5	Різноманітність методів монетизації	Модель підписки за обліковий запис не єдина можлива для проекту, можна виділити продаж доступу до курсів стороннім слухачам, додаткову розширену функціональність продавати як доповнення, розробка додаткових функцій під замовлення	Вибір моделі підписки на повну версію системи з ціною за 1 обліковий запис. Розробка окремої версії на базі існуючої системи для орієнтації на сторонніх слухачів. Розробка окремої функціональності під замовлення (модель продажів CMS систем).

Висновки: вибрані фактори можливостей говорять про те, що вихід проекту на ринок можливий, існує велика аудиторія, що зацікавлена в перевагах які пропонуються. Незважаючи на фактори загроз, вони можуть бути обернуті в можливості та принести додаткових клієнтів. Одним з важливих факторів можливостей є відкритість до монетизації – окрім обраного варіанту впровадження інших займатиме небагато часу та не вимагатиме додаткових інвестицій. Інші можливості явно показують, що вихід проекту на ринок сприятиме збуту бізнес-продукту та стрімкого зростання доходів як інвесторів так і власників, а в майбутньому, імовірно, викладачів.

9.7 Ступеневий аналіз конкуренції на ринку

Ступеневий аналіз конкуренції на ринку наведено в таблиці 9.7.

Таблиця 9.7 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
Тип конкуренції:	Монополістична.	Немає суттєвого впливу
	Локальний. З'являються дрібні конкуренти, які намагаються зайняти певний навчальний заклад або освітню мережу.	
	Національний. Присутній конкурент на національному ринку.	
	Інтернаціональний. Присутні конкуренти зі світовим іменем, локалізацією на понад 40 мов та охоптом аудиторії 40млн+ слухачів.	
За галузевою ознакою:	Внутрішньогалузева. Конкуренція відбувається в галузі освітніх наукових послуг.	Необхідно прикласти зусиль щоб впровадити функціональність, яка відсутня або недосконала в конкурентів, зорієнтуватися на викладача, чого немає в запропонованих досі проектах та платформах
	Товарно-родова. Існує конкуренція не тільки між онлайн-платформами а й з очними курсами, які представлені на ринку освітніх послуг.	
	Товарно-видова. Конкуренція з онлайн платформами навчання Prometheus, Coursera, Moodle, тощо	

	Між бажаннями. Не задекларовано конкуренції між бажаннями. Ринок немає розподілення на елітні та доступні системи.	компанії не можуть бути користувачами платформи, тому представляють собою конкурентів.
	Нецінова. При виборі платформи для онлайн-навчання споживач звертає увагу на відгуки, досвід та навички викладача, тематику курсу та відповідність сучасним потребам. Звертає увагу на зручність інтерфейсу та способи систематизації матеріалів.	
	Цінова. Для значної частки слухачів ціна є вагомим критерієм вибору.	
За інтенсивністю: (марочна, немарочна)	Немарочна. Для викладача та слухача більш важливі функціональність та наповнення ніж марка платформи.	Забезпечити наповнення курсів авторськими матеріалами від провідних викладачів. Це зробить систему більш конкурентноспроможною не тільки на локальному рівні а й на національному

Висновки: ступеневий аналіз показує, що внесення пропозиції на ринок є актуальним, а проект може конкурувати як з існуючими рішеннями на локальному рівні так і вийти на національний. Визначено, що ціновий фактор є вагомим, проте не вирішальним, а від впливу діяльності підприємства напряду залежать позиції продукту на ринку. Це означає, що правильно прийняті рішення та якісно надані послуги можуть забезпечити проекту довгострокове лідерство в галузі.

9.8 Аналіз конкуренції в галузі за м. Портером

З урахуванням аналізу конкуренції проведено більш детальний аналіз умов конкуренції в галузі за моделлю 5 сил Портера, дані аналізу наведено в таблиці 9.8.

Таблиця 9.8 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
		Бар'єри входження в ринок	Фактори сили постачальників	Фактори сили споживачів	Фактори загроз з боку замінників
	Moodle	Інертність цільової аудиторії	Ціна хостингу	Вимоги щодо зручності	Курси підвищення кваліфікації
	Coursera	Відсутність реальних вимог	Обмеження наповнення контентом	Активність	Очні курси за темами
	Prometheus	Складність інтеграції в навчальний процес	Підтримка технологій	Наповнення та робота з матеріалами	Програми стажувань від відомих локальних компаній
	Існуючі програми для проведення лабораторних робіт з комп'ютерної криптографії	Залежність від підтримки та вимоги щодо реакції на звернення	Доступ до технології	Ключові активи на ринку – це слухачі та викладачі	Відео лекції та завдання що перевіряються викладачами очно
Висновки	Інтенсивність конкуренції слабка з боку Coursera та Prometheus через різну орієнтованість. Інтенсивність конкуренції з боку існуючих програм для проведення лабораторних	Потенційні конкуренти відсутні або не анонсували планів щодо виходу на ринок. Вільний вхід на ринок можливий. Строки входу прив'язані до навчальних	Постачальники не впливають на роботу проєктів на ринку. Насиченість технологіями дозволяє не перейматися за обмеження доступу до технологій, що використовуються для	Клієнти диктують умови щодо зручності та зрозумілості інтерфейсу користувача. Напов-	Обмежень для роботи на ринку з боку товарів замінників немає. Можлива часткова стагнація пов'язана з сезонністю та

	робіт середня, пояснюється актуальним використанням у навчальному процесі проте компенсується низькою якістю продукту та відсутністю підтримки. Інтенсивна конкуренція з боку Moodle як більш відомого аналога проте з меншою орієнтованістю на локальний рівень, долається шляхом впровадження необхідних функцій і покращенням існуючих.	планів та графіків. Для входу потрібно подолати конкуренцію існуючих рішень та забезпечити функції системи, що стануть ключовими у прийнятті рішення про використання системи адміністрування електронних навчальних курсів. Потрібна підтримка сумісності з технікою закладу або хмарність.	реалізації рішень. Обрані технології реалізації вже мають велику кількість активних користувачів, що забезпечують їм широку спільноту та колективне вирішення проблем.	ненням курсів займаються клієнти-викладачі від якості їх матеріалів залежить якість наданих слухачам послуг щодо онлайн освіти. За необхідності клієнти можуть вимагати додаткових функцій у системі	прив'язаністю до графіків закладів освіти, канікул.
--	--	--	--	--	---

Висновки: з огляду на конкурентну ситуацію висновок щодо принципової можливості роботи на ринку наступний:

- робота на ринку можлива та рекомендована;
- потрібно швидко реагувати на запити клієнтів, оскільки від них залежить успішність проекту;
- через використання сильних сторін проекту необхідно займати локальний рівень, надавати якісні послуги, розширювати спектр функцій та починати активне впровадження системи на національному рівні;
- проект має бути розроблено з врахуванням можливості швидкого переходу на технології іншого постачальника або зміни хостінг провайдера.

До факторів конкурентоспроможності варто віднести наступні:

- відсутність конкуренції у галузі систем орієнтованих на викладача;
- зайняття цієї ніші дозволить посилити вплив на частині ринку орієнтованій на слухачів;
- якісні сервіси, що відповідають описаним функціям забезпечать високий рівень задоволеності клієнтів;
- функції вибрані для обґрунтування факторів конкурентоспроможності є унікальними або суттєво покращують досвід взаємодії з системами адміністрування електронних навчальних курсів.

9.9 Обґрунтування факторів конкурентоспроможності

Фактори конкурентоспроможності вибрані на основі аналізу конкуренції, врахування характеристик ідеї проекту, вимог споживачів до товару та факторів маркетингового середовища такі:

- зручність у використанні;
- комплексний склад оцінки;
- зворотній зв'язок;
- автоматизація рутинних процесів;
- простота інтеграції;
- засіб покращення існуючої форми навчання а не заміна;
- розширення через розробку;
- орієнтованість на викладача.

Обґрунтування факторів конкурентоспроможності наведено в таблиці 9.9.

Таблиця 9.9 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (чинники, що роблять фактор для порівняння конкурентних проектів значущим)
1	Зручність у використанні	Інтерфейс системи – те з чим взаємодіє користувач напряду, повинен бути простим, зрозумілим, зручним а система забезпечувати виконання заявлених функцій
2	Комплексний склад оцінки	Слухачі очікують отримувати оцінки, що базується не тільки на проходженні тесту а з врахуванням зусиль які вони вклали для вивчення теми або проходження курсу (в т.ч. ознайомлення з додатковими матеріалами та першоджерелами)
3	Зворотній зв'язок	Прогрес у проходженні курсу повинен відображатися для того, щоб розраховувати наскільки просувається робота
4	Автоматизація рутинних процесів	Викладачі утримуються від інтеграцій систем адміністрації електронних навчальних курсів через велику кількість рутинних процесів – наповнення, реєстрація користувачів, невідповідність структури програми структурі навчального закладу
5	Простота інтеграції	Розгортання системи для нового клієнта не повинне займати багато часу, система повинна бути доступною для слухачів незалежно від машини на якій вони працюють
6	Засіб покращення існуючої форми навчання а не заміна	Онлайн-курси призначені замінити традиційні методи навчання та денну форму, реальні вимоги викладачів вказують на те, що потрібен інструмент який зможе допомогти проводити очні заняття та слідкувати за активністю, виконанням завдань, дотриманням графіку слухачами які за тих чи інших причин не були присутні

7	Розширення через розробку	У випадку розширення через розробку додаткових курсів система збирає навколо себе спільноту зацікавлених технічно підкованих людей, що можуть принести додаткові функції, через це відбудеться тісна інтеграція з сервісами системи і збереження позицій на ринку стане простішим.
8	Орієнтованість на викладача	Викладач – повноцінний користувач системи з власними потребами та вимогами. Наповнення курсів, аналітика, автоматизація рутинних процесів обов'язкові для зручної роботи з платформою.

Висновки: визначено та обгрунтовано справедливість використання факторів конкурентоспроможності на основі яких може і буде проведено порівняльну оцінку прямих конкурентів та запропонованого проекту. Проте вже на цьому етапі аналізу, очевидно, що проект є повністю конкуренто-спроможним і відповідає вимогам для входу в ринок.

9.10 Порівняльний аналіз сильних та слабких сторін

Порівняльний аналіз системи адміністрування електронних навчальних курсів та прямих конкурентів наведено в таблиці 9.10.

Таблиця 9.10 - Порівняльний аналіз сильних та слабких сторін «САЕНК»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з «САЕНК»						
			-3	-2	-1	0	+1	+2	+3
1	Зручність у використанні	10					+		
2	Комплексний склад оцінки	18				+			
3	Зворотній зв'язок	15				+			

4	Автоматизація рутинних процесів	18		+					
5	Простота інтеграції	15		+					
6	Засіб покращення існуючої форми навчання а не заміна	20		+					
7	Розширення через розробку	20		+					
8	Орієнтованість на викладача	20			+				

Висновки: порівняльний аналіз наочно підтверджує припущення зроблене раніше про конкурентоспроможність проекту та доцільність входу в ринок. Серед недоліків варто відзначити зручність у використанні, конкуренти роками досліджували ринок та вносили покращення в інтерфейс користувача, що ставить їх у вигравне становище.

9.11 Swot аналіз стартап-проекту

SWOT аналіз наведено в таблиці 9.11.

Таблиця 9.11 – SWOT аналіз стартап-проекту

<p>Сильні сторони: Простота розгортання та розширення за рахунок сторонніх застосунків. Автоматизація процесів оцінювання, реєстрація великої кількості нових слухачів, формування файлів з даними, відображення активності слухачів у відносній формі. Викладач конфігурує склад оцінки за курс.</p>	<p>Слабкі сторони: Інтерфейс користувача. Зручність деяких функцій.</p>
<p>Можливості: Можливі вдосконалення та розширення проекту за рахунок зміни орієнтації з викладача на слухача та внесення відповідних змін.</p>	<p>Загрози: Компанії конкуренти можуть швидко впровадити подібну функціональність і інтегрувати її в системи через</p>

Вдосконалення інтерфейсу взаємодії з користувачем.	оновлення, таким чином вони покривають значну долю ринку
--	--

На основі SWOT-аналізу розроблено альтернативи ринкової поведінки для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок.

Альтернативи ринкової поведінки перелічено в таблиці 9.12.

Таблиця 9.12 – Альтернативи ринкової поведінки

№ п/п	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Підписка з вартістю за обліковий запис викладача (1 демо аккаунт)	90%	2 міс.
2	Прямі продажі	50%	3 міс.
3	Персоналізована реклама	30%	3 міс.

Обрана модель №1 - поширення за підпискою з оплатою за додаткові облікові записи викладачів. Демо обліковий запис у якості маркетингового ходу дозволить поширити систему та продемонструвати функціональність широкому колу зацікавлених потенційних клієнтів. Вибір обґрунтовується найбільшою з запропонованих імовірністю отримання ресурсів 90% та найбільш стиснутими строками реалізації – 2 місяці.

9.12 Розроблення маркетингової програми стартап-проекту

Підсумок результатів аналізу конкурентоспроможності товару наведено в таблиці 9.13.

Таблиця 9.13 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1	Використання програмного комплексу для спрощення процесу оцінювання	Викладач чітко може відслідковувати активність слухачів, бачити результати не тільки в абсолютних цифрах а й у відносних	Аналогічної системи моніторингу досі не використано в жодного конкурента
2	Можливість розширення без притягнення до розробки компанії-постачальника	Забезпечує механізм розширення через невтручання. Засоби розширення розробляються у комплексі, що має бути інтегрований	Подібна функціональність є в Moodle, проте впровадження без участі адміністратора комплексу неможливе
3	Автоматизація рутинних процесів	Викладач має змогу отримати рекомендовані оцінки, створити аккаунти, підтягнути дані зі стороннього комплексу зі значною економією часу	Журнал оцінок в Moodle дозволяє імпортувати результати, проте система надає результати в найбільш звичному викладачу форматі

Трирівневу маркетингову модель товару, уточнену ідею та характеристики наведено в таблиці 9.14.

Таблиця 9.14 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові	
I. Товар за задумом	Задоволення потреби в оновленні програмних комплексів лабораторних робіт, для студентів денної форми навчання. Необхідність у панелі адміністрування курсів, автоматизації рутинних завдань викладача.	
	Властивості, характеристики	Значення
	Складові	Клієнтський веб-застосунок, серверний застосунок, база даних MS SQL Server
	Взаємодія	API

	Кількість одночасних звернень (за час обробки 1 запиту)	20000
	Простота набуття навичок	Підказки та візуальні ефекти
	Зручність у використанні	Елементи керування розміщені очевидно
	Підтримка на час впровадження	
	Реакція на знайдені помилки, виправлення протягом року	
За рахунок чого потенційний товар буде захищено від копіювання	Вихідний код продукту буде опубліковано на Github, із моменту публікації – програма захищається відповідно до закону України про авторське право і суміжні права	

Визначення меж ціни наведено в таблиці 9.15.

Таблиця 9.15 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	1900 грн (24 грн / \$)	Немає	11000 – 80000 грн	100 – 500 грн/о.з. 100-5000 у.о. за доп. функції

Визначення оптимальної системи збуту наведено в таблиці 9.16.

Таблиця 9.16 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Покупка облікового запису викладача через підписку	Розгортання системи, надання даних для доступу, початковий інструктаж, підтримка	Локальний рівень	Без посередників, рекламні майданчики Facebook та Google

Розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів наведено в таблиці 9.17.

Таблиця 9.17 – Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Реклама повинна розміщуватися на ресурсах, пов'язаних з освітою та ІТ сферою. Допустиме розміщення на зтематичних заходах, оскільки цільова група є частими відвідувачами таких заходів та ресурсів	Електронна пошта, Telegram, Viber	Лідер у системі трекінгу активності слухача. Пропрацьована систематизація матеріалів. Продукт економить час на рутинних діях.	Донести інформацію про те, що продукт дозволить оцінювати студентів справедливо та організовувати навчальний курс більш систематизовано. Акцентувати увагу на сучасності онлайн-освіти.	Короткі повідомлення орієнтовані на цільову аудиторію. Демо на 5 хвилин із роботою системи.

Результатом проведеного аналізу є вибір стратегії конкурентної поведінки: стратегія заняття конкурентної ніші. Та розроблено стратегію маркетингу – введення у свідомість цільової групи повідомлень про необхідність збереження часу, шляхом впровадження системи, сучасність та популярність онлайн-рішень у студентів і школярів. Визначено ключові канали комунікації з цільовою аудиторією.

Висновки: вхід на ринок є обґрунтованим, враховуючи показники аналізу та висновки зроблені раніше – проект є рентабельним та конкурентоспроможним. Незважаючи на велику кількість ризиків у випадку правильної рекції компанії-

постачальника продукту ці ризики нівелюються та переходять в категорію можливостей. Ринок не є перенасичений конкурентами а тому швидке впровадження в обрану нішу дозволить подолати в конкурентній боротьбі більш імениті аналоги. Окрім того на ринку не прослідковується прямої боротьби за кінцевого слухача, оскільки він може бути користувачем кількох платформ. Обрана альтернатива ринкової поведінки дозволяє розраховувати на швидке впровадження у навчальний процес і майбутній дохід від розголосу у кругах потенційно зацікавлених клієнтів. Перспективи проекту також чітко прослідковуються за рахунок гнучкості та запропонованої функціональності дозволить довго утримувати зайняті позиції. Розробка, покращення та підтримка проекту в подальшому – доцільні, оскільки дозволять швидшими темпами заволодіти частиною ринку та продемонструвати клієнтам про серйозні наміри залишитися в галузі.

ВИСНОВКИ

Метою було проектування та розробка першої версії системи адміністрування електронних навчальних курсів, яка вирішить проблему керування кількома розрізненими лабораторними практикумами а також підтримуватиме можливість створення вбудованих простих електронних курсів.

В результаті досягнуто поставленої мети. Розроблено першу версію системи, що включає функції, які було вибрано критеріями покращення та такими, що досі не були включені в систему подібного типу за рахунок чого частково вирішену поставлену проблему.

В ході виконання робіт проаналізовано існуючі рішення, вибрано критерії для покращення та функції, які дозволяють вирішити проблему, які досі не були розроблені. Обґрунтовано ефективність реалізації цих функцій та сформовано вимоги до системи. Описано всі необхідні дані та структури тощо.

На основі вимог розроблено діаграму прецедентів і описано кожен варіант використання детально. Крім того проведено розділення варіантів використання поміж типами користувачів у системі та детально описано різницю між правами різних користувачів. Використано даний опис для вибору технологій, які дозволять повною мірою реалізувати систему та спростять розробку. Вибір технологій зроблено шляхом аналізу переваг та недоліків кожної з технологій. Крім того, кожна наступна вибрана технологія також повинна була підтримувати попередню та працювати з нею без необхідності написання певних заглушок. Таким чином, завдяки аналізу та порівнянню вибрано стек технологій, що безперешкодно взаємодіють одна з одною, виконують необхідні функції та дозволяють спростити розробку систему, в кінцевому рахунку.

Враховуючи вибрані технології визначено необхідні модулі системи, та спроектовано структурну схему системи, детально описано її складові та представлено їх внутрішню структуру. Використано популярні практики проектування подібних застосувань. Після оцінки напрацювань та з використанням сформованих вимог спроектовано спочатку модель БД, з усіма зв'язками та

важливими атрибутами, визначено обмеження та на основі моделі розроблено базу даних. Результати цієї роботи представлено на рисунках та в додатках. Під час розробки, використано ORM, що означає спрощення проектування доменних сутностей при наявності схеми БД і готових зв'язків між таблицями.

З урахуванням вищенаведених досягнень розроблено загальну архітектуру системи, при цьому використано як сформовані вимоги та загальну структурну схему а й найкращі відомі практики. Вирішено проблеми відстеження активності слухача, використано можливості мови програмування С# та, зокрема фреймворку ASP.NET Core для пришвидшення та спрощення розробки, виділено загальні архітектурні елементи в окремі бібліотеки, що можуть бути використані та розширені незалежно від усієї системи. Спроектовано та розроблено API системи, чим вирішено поставлені завдання. Розроблено інтерфейс користувача для демонстрації роботи системи.

Розроблено стартап проект у якому проаналізовано ринки збуту, обрано цільову аудиторію та проведено порівняння з існуючими аналогами з точки зору бізнесу. Враховуючи всі етапи проведеного аналізу вибрано альтернативу ринкової конкурентної поведінки, ключові маркетингові точки, що за всіма підрахунками та зробленими висновками забезпечить стабільний розвиток проекту як стартапу протягом перших фаз життєвого циклу продукту.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Полтораки В.П. Теорія інформації та кодування / Ю.П. Жураковський, В.П. Полтораки // Підручник. – К.: Вища школа, 2001. – 255 с.: іл., укр.мовою (Гриф МОНУ №1/11-2367 від 21.05.2001)."
2. Маркозов Д. О. ДИСТАНЦІЙНЕ НАВЧАННЯ ЯК ІННОВАЦІЙНА ФОРМА ОСВІТИ [Електронний ресурс] / Д. О. Маркозов. – 2009. – Режим доступу до ресурсу: http://www.rusnauka.com/24_PNR_2009/Pedagogica/50819.doc.htm.
3. Офіційний сайт ArtMoney [Електронний ресурс] – Режим доступу до ресурсу: <http://www.artmoney.ru/manual/english/help.htm>.
4. About Cheat Engine [Електронний ресурс] – Режим доступу до ресурсу: <https://www.cheatengine.org/aboutce.php>.
5. Moodle statistics [Електронний ресурс] – Режим доступу до ресурсу: <https://stats.moodle.org/>.
6. Про проект – Prometheus [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://prometheus.org.ua/about-us/>.
7. About | Coursera Blog [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://about.coursera.org/>.
8. Полтораки В.П. Криптографічний захист даних в цифрових інформаційних системах (Частина 1) / Полтораки В.П. // Телеком. Військовий зв'язок. Спеціальний випуск, №2/2018. - К.: Softpress. hi-Tech.ua, Жовт., 2018. - с. 98-104., Мова публікації:українська
9. Operating System Market Share Worldwide | StatCounter Global Stats [Електронний ресурс] – Режим доступу до ресурсу: <https://gs.statcounter.com/os-market-share>.
10. Брацький В. О. Дослідження особливостей застосування реляційних і нереляційних баз даних на прикладі SQL Server та MongoDB / В. О. Брацький, О. М. М'якишило // Наукові праці Національного університету харчових технологій. - 2016. - Т. 22, № 5. - С. 15-24. - Режим доступу: http://nbuv.gov.ua/UJRN/Npnukht_2016_22_5_4.

11. Старіков В. А. ОСОБЛИВОСТІ РОЗРОБКИ МОБІЛЬНОГО ДОДАТКУ З ВИКОРИСТАННЯМ ХМАРНОЇ БАЗИ ДАНИХ / В. А. Старіков. // "Молодий вчений". – 2017. – №11. – С. 1078–1080.
12. Pilov J. Understanding Optimizer Timeout and how Complex queries can be Affected in SQL Server [Електронний ресурс] / Joseph Pilov. – 2018. – Режим доступу до ресурсу: <https://blogs.msdn.microsoft.com/psssql/2018/10/19/understanding-optimizer-timeout-and-how-complex-queries-can-be-affected-in-sql-server/>.
13. Allen E. B. Measuring coupling and cohesion: an information-theory approach [Електронний ресурс] / E. B. Allen, T. M. Khoshgoftaar // IEEE. – 2002. – Режим доступу до ресурсу: <https://ieeexplore.ieee.org/abstract/document/809733>.
14. Tempero E. What programmers do with inheritance in Java [Електронний ресурс] / E. Tempero, H. Yang, J. Noble. – 2013. – Режим доступу до ресурсу: <https://intellect.icu/pochemu-nuzhno-predpochitat-kompozitsiyu-pered-nasledovaniem-7920>.
15. Harvie L. 4 Ways to Make Your Code More Reusable [Електронний ресурс] / Lance Harvie. – 2018. – Режим доступу до ресурсу: <https://medium.com/@lanceharvieruntime/4-ways-to-make-your-code-more-reusable-bc20889c1e4>.
16. Weiss T. R. Microservices Pricing: What's It All Going to Cost? [Електронний ресурс] / Todd R. Weiss. – 2018. – Режим доступу до ресурсу: <https://thenewstack.io/microservices-pricing-whats-it-all-going-to-cost/>.
17. Chwedczuk M. How NOT to use the repository pattern [Електронний ресурс] / Marcin Chwedczuk. – 2018. – Режим доступу до ресурсу: <https://marcin-chwedczuk.github.io/repository-pattern-my-way>.
18. Arh D. How does a C# LINQ Query execute and How does this affect Performance? [Електронний ресурс] / Damir Arh. – 2019. – Режим доступу до ресурсу: <https://www.dotnetcurry.com/csharp/1481/linq-query-execution-performance>.

ДОДАТОК 1 – Діаграма прецедентів

ДОДАТОК 2 – Структурна схема системи

ДОДАТОК 3 – Інфологічна модель

ДОДАТОК 4 – Діаграма БД

ДОДАТОК 5 – Діаграма розгортання

ДОДАТОК 6 – Архітектура бібліотеки роботи з Excel

ДОДАТОК 7 – Загальна архітектура системи

ДОДАТОК 8 – Діаграма послідовності імпорту стороннього курсу